DeepAVP: A Dual-Channel Deep Neural Network for Identifying Variable-Length Antiviral Peptides

Jiawei Li, Yuqian Pu, Jijun Tang, Quan Zou, and Fei Guo

***Abstract*—Antiviral peptides (AVPs) have been experimentally verified to block virus into host cells, which have antiviral activity with decapeptide amide. Therefore, utilization of experimentally validated antiviral peptides is a potential alternative strategy for targeting medically important viruses. In this article, we propose a dual-channel deep neural network ensemble method for analyzing variable-length antiviral peptides. The LSTM channel can capture long-term dependencies for effectively studying original variable-length sequence data. The CONV channel can build dynamic neural network for analyzing the local evolution information. Also, our model can fine-tune the substitution matrix for specifically functional peptides. Applying it to a novel experimentally verified dataset, our AVPs predictor, DeepAVP, demonstrates state-of-the-art performance of 92.4% accuracy and 0.85 MCC, which is far better than existing prediction methods for identifying antiviral peptides. Therefore, DeepAVP, web server for predicting the effective AVPs, would make significantly contributions to peptide-based antiviral research.**

***Index Terms*—Antiviral peptides, dual-channel deep neural network, web sever.**

I. Introduction

ANTIVIRAL peptides (AVPs) have been experimentally verified to block virus attachment or the entry of a virus into host cells [1], [2]. It is just possible that the antiviral peptides may interfere with key steps that a pathogenic mammalian virus needs to enter a cell. Antiviral peptides are substantially identical to a small portion of a glycoprotein in the virus, which have antiviral activity against influenza virus with the decapeptide amide. Over the past decade, the antiviral research has always been a considerable focus of scientists [3].

Due to the limited availability of therapeutic molecules for many viral infections, we need to explore new antiviral candidates to control pathogenic re-emerging and resistant viruses [4]. Therefore, experimentally validated antiviral peptides can be used as a potential alternative strategy for targeting medically important viruses [5], [6]. In recent years, AVPs prediction tools collected and predicted highly effective antiviral peptides via traditional machine learning algorithms. The first AVPs prediction tool, AVPpred [7], is the collection and prediction of highly effective antiviral peptides via traditional machine learning algorithm. Chang KY *et al.* [8] demonstrated that a physicochemical model using random forests outperform in distinguishing antiviral peptides. Zare1 M *et al.* [9] studied the concept of pseudo-amino acid composition (PseAAC) and utilized Adaboost to classify antiviral peptides. AntiVPP 1.0 [10] used the Random Forest algorithm for antiviral peptide predictions, via net charge, number of hydrogen bond donors, molecular weight and hydropathy index.

Many studies report highly efficient peptides against human viruses, e.g. influenza [11], [12], HIV [13], WNV [14], HCV [15], HSV [16], RSV [17] and etc. Several naturally occurring antimicrobial peptides [18]–[20] have been made in an attempt to identify important functions further contributing to the antiviral activity, such as Thomas *et al.* [21], Wang *et al.* [22], and some other general antimicrobial peptide prediction tools [23], [24].

Recently, the learning technology is especially formidable in handling mass biomedical data and achieves great success in a wide variety of bioinformatics applications [25]–[31]. With the advances of big data era in biology, it is foreseeable that deep learning method becomes increasingly important in the field of proteomics [32]–[34]. A small amount of deep neural network models, including the convolutional and recurrent layers that leverage primary sequence composition, are utilized to recognize various types of peptides as follows, Veltri [35] for antimicrobial peptides and Bulik [36] for HLA peptides.

In this paper, we propose a computational method based on deep neural network for predicting antiviral peptides. Our model is a dual-channel deep neural network, in order to extract different dimensional features from original variable-length

Manuscript received November 6, 2019; revised February 11, 2020; accepted February 24, 2020. Date of publication February 28, 2020; date of current version October 5, 2020. This work was supported by a grant from National Natural Science Foundation of China under Grants (NSFC 61772362, 61972280) and National Key R&D Program of China under Grants 2018YFC0910405, 2017YFC0908400. *(Corresponding authors: Quan Zou; Fei Guo.)*

Jiawei Li, Yuqian Pu, and Fei Guo are with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: jiawei6636@tju.edu.cn; yqianpu@tju.edu.cn; fguo@tju.edu.cn).

Jijun Tang is with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China, and with the Key Laboratory of Systems Bioengineering (Ministry of Education), Tianjin University, Tianjin 300072, China, and also with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: tangjijun@tju.edu.cn).

Quan Zou is with the Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: zouquan@nclab.net).

Digital Object Identifier 10.1109/JBHI.2020.2977091

2168-2194 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

sequence data. The LSTM module imports the peptide sequence length as an important element to classification the antiviral peptides, and the bi-directional recurrent neural network (BLSTM) can capture long-term dependency for effectively studying sequence data. The CONV module applies the substitution matrix as kernels to extract the convolutional features, and the dynamic neural network can fine-tune the substitution matrix for specifically functional peptides. The final joint module concatenates the LSTM and CONV channels by two fully-connected layers, which integrates evidence to classify the antiviral peptides.

Important characteristic of our model is that we processe sequence data with no need for feature extraction, whereas the LSTM and CONV channels can analyze peptide sequences from sequential and evolutionary levels, respectively. Even more, the input of our model is variable length sequence, which is just a peptide with any length range from several residues to hundred or thousand residues that has great scalability. In the LSTM channel, we use the state output with time step specific to sequence length. In the CONV channel, we add AvBlock layer to do average block on the sequence length PSSM matrix. Our predictive model has several key competitive advantages rather than other outstanding prediction tools.

II. Materials and Methods

In this study, we analyze variable-length antiviral peptides via a dual-channel deep neural network ensemble method. On the one hand, we design a bi-directional recurrent neural network that extracts the sequence features from one-hot encoding. On the other hand, we propose a dynamic convolutional neural network that extracts the evolution features from amino acid substitution matrix. Final, we construct a dual-channel connection model that integrates the evidence to identify antiviral peptides. As shown in Fig. 1, we concatenate the LSTM and CONV channels by two fully-connected layers.

A. Data Set

In our study, we focus on the recognition of antiviral peptides. Here, we evaluate on the well-established dataset, which is proposed by Nishant *et al.* [6]. The peptide sequences are collected with a reported antiviral activity against human viruses like HIV, HCV, SARS and Influenza, etc. More than 90% of antiviral peptides are extracted from natural source, and remaining peptides have synthetic source. Therefore, 604 highly effective antiviral peptides and 452 least or non-effective antiviral peptides have been processed by Nishant *et al.* [6], as one training set $T^{544p+407n}$ (544 positive and 407 negative) and one testing set $V^{60p+45n}$ (60 positive and 45 negative). Also, they have taken non-experimental negative peptides, as one training set $T^{544p+544n*}$ and one testing set $V^{60p+60n*}$. The negative peptides have been employed in earlier antimicrobial peptide prediction method [37]. On the real dataset, the positive and negative samples are all extracted by experimental technology. However, on the real dataset, the positive samples are extracted by experimental technology and the negative samples are collected from existing database. The length distributions of two benchmark datasets are shown in Fig. 2. On the real dataset, most of positive peptides are always much shorter than a lot of negative peptides. However, on the random dataset, the length distributions between the positive and negative peptides are exactly similar.

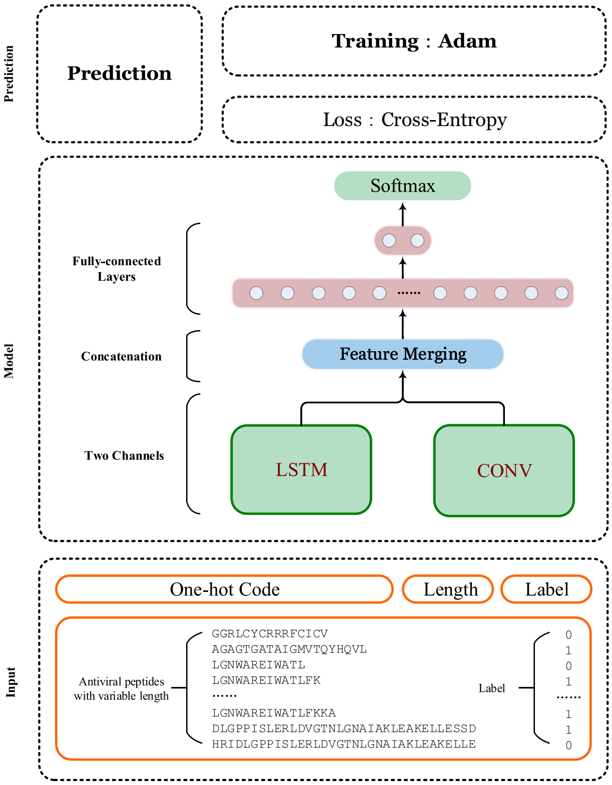

Fig. 1. DeepAVP: deep learning model for identifying variable-length antiviral peptides.

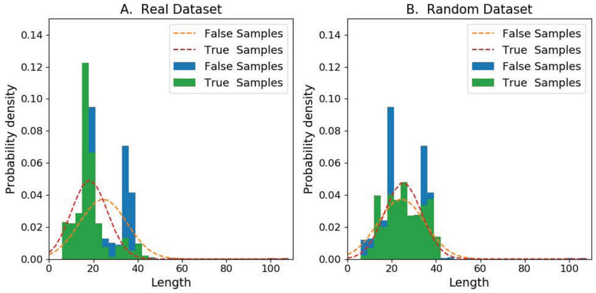

Fig. 2. Length distributions of two benchmark datasets.

With the development of antiviral peptides (AVPs) research [3], [38], [39], in addition to previous dataset, the AVPs databases have emerged in large numbers. To update and expand the AVPs dataset, we extract 916 highly effective antiviral peptides from four different datasets (AVPdb [40], APD3 [41], CAMPR3 [42], LAMP [43]) and 452 non-effective antiviral peptides from one database (AVPdb). The homologous sequences are removed by CD-hit [44] if they shared a high sequence identity (greater than 90%) with any sequence in the dataset. Finally, we obtain 413 AVPs and 348 non-AVPs as a novel non-redundant AVPs dataset.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

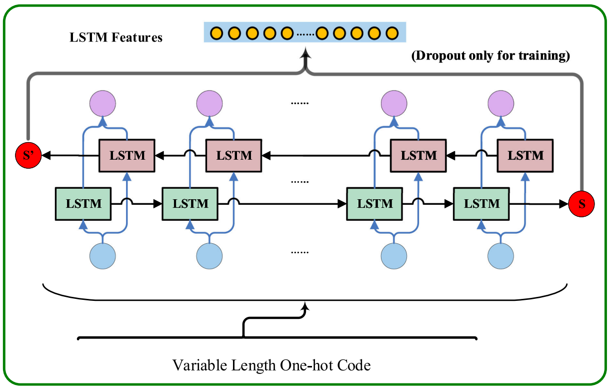

Fig. 3. The variable-length bi-directional LSTM channel.

B. LSTM Module

The LSTM module is one bi-directional recurrent neural network to deal with one-hot encoding, as shown in Fig. 3.

1) Sequence Representation: The one-hot encoding is a group of bits among which the legal combinations of values are only those with a single high “1” and all others low “0”. Each peptide sample consists of one L-length amino acid sequence, which can be one-hot encoded into the $L \times 20$ binary matrix, with the column corresponding to 20 amino acid types.

2) Recurrent Neural Network: Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture [45], where all connections between units form a directed cycle. It creates an internal state of the network that allows to exhibit dynamic temporal or spatial behavior.

Here, we build a bi-directional long short-term memory network (BLSTM), which is a variant of RNN that combines the outputs of two RNNs, one processing the sequence from left to right, the other one from right to left. Two RNNs contain some LSTM blocks, which can remember a value for an arbitrary length of sequence data. We regard the input sequence length as the number of time steps, and get the output of final time step in two directions.

The unit of LSTM is dynamically adjusted by the input sequence length. Each LSTM unit is comprised of the input gate, the forget gate and the output gate, the formulation can be expressed as follows:

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f b_{t-1} + b_f)\\
i_t &= \sigma(W_i x_t + U_i b_{t-1} + b_i)\\
o_t &= \sigma(W_o x_t + U_o b_{t-1} + b_o)\\
C_t &= i_t \circ \tanh(W_c x_t + U_c b_{t-1} + b_c) + f_t \circ C_{t-1}\\
h_t &= o_t \circ \tanh(C_t)
\end{aligned} \tag{1}$$

where x_t is input vector, f_t is forget gate’s activation vector, i_t is input gate’s activation vector, o_t is output gate’s activation vector, h_t is hidden state vector, C_t is cell state vector, W and U are parameter matrices and b is a bias vector.

In the LSTM channel, we use the state output with time step specific to sequence length. Here, we select h_t as the output with 128 dimensions. Also, we add a dropout layer after the output to prevent over-fitting, as setting keep-prob as 0.8.

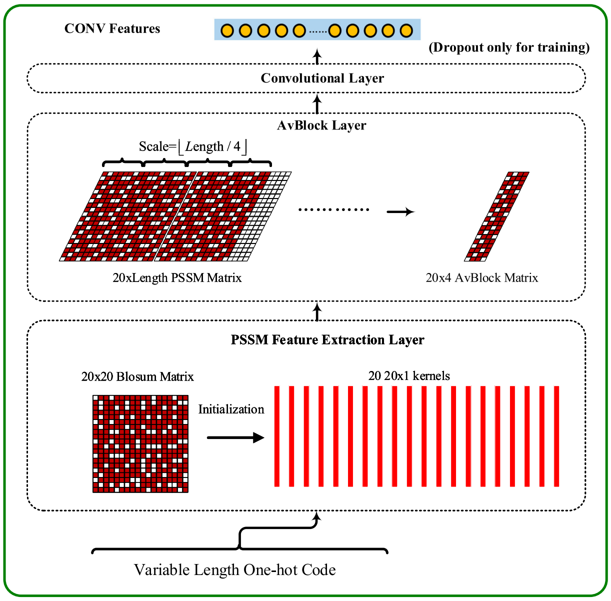

Fig. 4. The variable-length dynamic CONV channel.

C. CONV Module

The CONV module is a dynamic convolutional neural network to deal with position specific scoring matrix, as shown in Fig. 4.

1) Evolution Representation: The position specific scoring matrix (PSSM) is a commonly used representation to reflect evolution information in biological sequences [46]. For one L-length sequence, position specific scoring matrix can be denoted by the $L \times 20$ value matrix as follows:

$$PSSM = \begin{pmatrix} p_{1,1} & \cdots & p_{1,j} & \cdots & p_{1,20}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ p_{i,1} & \cdots & p_{i,j} & \cdots & p_{i,20}\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ p_{L,1} & \cdots & p_{L,j} & \cdots & p_{L,20} \end{pmatrix} \tag{2}$$

where $p_{i,j}$ stands for the score that amino acid in i-th position being turned into the j-th type during the evolution process.

2) Convolutional Neural Network: Convolutional neural network (CNN) is designed to extract features from high-dimensional data, while keeping the number of model parameters tractable by applying a series of convolutional and pooling operations.

Here, we build a dynamic convolutional neural network (DCONV), which is a variant of CNN that the average block (AvBlock) model is embedded into the convolutional neural network. It is comprised of the feature extraction layer, the AvBlock layer and the convolutional layer.

For the feature extraction layer, we initialize twenty 20×1 convolution kernels by using the 20×20 BLOSUM matrix [47].

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: AvBlock Model.

```
function AvBlock($tensor\_list, width$)
    while $t_i \in tensor\_list$ do
        $block = length(t_i)/width$
        for $s = 1 : width$ do
            $nt_i(s) = avg\{t_i[block \times (s-1) + 1, block \times s]\}$
        end for
        $new\_tensor\_list.add(nt_i)$
    end while
    return $new\_tensor\_list$
end function
```

We may not use traditional technology [48], but build neural network model to construct position specific scoring matrix, the formulation can be expressed as follows:

$$Evo(X)_{i,k} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{m,n}^{k} X_{i+m,n} \tag{3}$$

where X is the input data, i is the index of positions and k is the index of kernels. Each convolution kernel W^k is an $M \times N$ weight matrix with M being the window size and N being the number of input channel, where $M = 1$ and $N = 20$. In addition, fine-tuned BLOSUM matrix can be trained out that differs from original BLOSUM matrix.

For the AvBlock layer, we make the variable evolution data into fixed-length features. The AvBlock model can dynamically produce the block according to input length, and compute the average in each block, as shown in Algorithm 1.

For the convolutional layer, we apply a convolution operation to above blocks, the formulation can be expressed as follows:

$$ConV(X)_{i,k} = ReLU\left\{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W_{m,n}^{k} X_{i*step+m,n}\right\} \tag{4}$$

where $M = 4$ and $N = 4$, the $step$ equals to 4, and the $ReLU$ represents rectified linear function as follows:

$$ReLU(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{5}$$

In the CONV channel, we use the AvBlock layer with space specific to sequence length. Here, we obtain the output with 100 dimensions. Also, we add a dropout layer after the output to prevent over-fitting, as setting keep-prob as 0.8.

D. Joint Module

The joint module takes concatenated last hidden vectors as input, and constructs two fully-connected layers and one softmax layer to identify antiviral peptides.

For two fully-connected layers with 100 and 2 neurons, the logits can come about by nonlinear combination of sequence and evaluation features. The formulation can be expressed as follows:

$$fulCN(X) = ReLU(WX + b) \tag{6}$$

where X is the features vector, W is a $M \times N$ weight matrix and b is a N-dimension bias vector.

For one softmax layer, the un-normalized vector can be normalized into a probability distribution. The standard (unit) softmax function is given by standard exponential function on each coordinate, divided by sum of all coordinates as a normalizing constant. The formulation can be expressed as follows:

$$softmax(X_i) = \frac{\exp X_i}{\sum_{k=0}^{K} \exp X_k} \tag{7}$$

where X represents given logits vector, i is the index of positions and K equals to 2, output coordinates sum to 1.

Final, we identify most likely choice with maximum prediction probability as the prediction label.

E. Model Training and Validation

We split the training set via five folds, and select one fold as the validation set. Our novel deep learning model is trained on four folds and verified on one fold, in order to save the best model on the validation set. We calculate the softmax cross entropy as loss function, and perform the Adam algorithm [49] to minimize loss function and optimize the model. Here, we set $learning\_rate$ as 0.01. Therefore, our model is fitted on the four-fold training set, hyper-parameters are optimized on the one-fold validation set, and the final performance and interpretation are exclusively reported on the test set.

III. Results and Discussion

In this section, we employ three benchmark datasets to evaluate our deep learning method. First, we analyze the performance of different modules to test the robustness of our method. Then, our method is compared with other outstanding methods under the independent test on well-established dataset. Finally, we analyze fine-tuned BLOSUM matrix that differs from original matrix in some details.

A. Evaluation Criteria

Specificity (SP), Sensitivity (SN), Accuracy (ACC) and Matthew's correlation coefficient (MCC) are employed to evaluate the performance of our method. They are calculated as follows:

$$\begin{aligned} SN &= \frac{TP}{TP+FN} \times 100 \\ SP &= \frac{TN}{TN+FP} \times 100 \\ ACC &= \frac{TP+TN}{TP+FP+TN+FN} \times 100 \\ MCC &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \end{aligned} \tag{8}$$

where S_n reflects the sensitivity, S_p reflects the specificity, A_c reflects the accuracy and MCC is the Mathew's correlation coefficient; while TP represents the true positive, TN represents the true negative, FP represents the false positive and

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

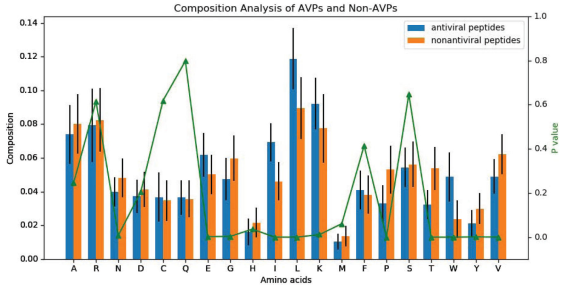

Fig. 5. Composition analysis of AVPs and Non-AVPs.

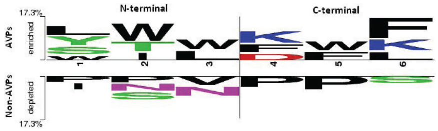

Fig. 6. Positional conservation Logo of three residues at N-terminal and C-terminals in AVPs against Non-AVPs.

TABLE I
PERFORMANCE OF THREE LSTM MODELS ON THE REAL DATASET

Data set	Model	SN	SP	ACC	MCC
$T^{544p+407n}$	UnidLSTM	86.4%	81.3%	84.2%	0.68
	BidLSTM	85.8%	80.4%	83.5%	0.66
	MultiLSTM	85.5%	79.6%	83.0%	0.65
$T^{60p+45n}$	UnidLSTM	90.0%	71.1%	81.9%	0.63
	BidLSTM	85.0%	86.7%	85.7%	0.71
	MultiLSTM	88.3%	73.3%	81.9%	0.63

FN represents the false negative. S_n, S_p and A_c stand for the success rates of prediction on positive, negative and overall datasets, respectively. MCC is used to evaluate the performance of predictor when the positive and negative samples in the dataset are out-of-balance. Its value ranges from 0 to 1 and a larger MCC means a better prediction.

In addition, Area Under Receiver Operating Characteristic (ROC) curve (AUC) and Area Under Precision Recall (PR) curve (AUPR) are used to evaluate the performance of our method. AUC is the area under receiver operating characteristic (ROC) curve, which is created by plotting true positive rate against false positive rate at various threshold settings. AUPR is the area under curve that is created by plotting precision against recall at various threshold settings.

B. Composition Analysis

To carry out the composition analysis of AVPs and Non-AVPs, we calculate the frequencies of all amino acids in the positive and negative datasets, as shown in Fig. 5. The analysis of AVPs reveals higher abundance of E, I, L, K and W (Welch's t-test, p value less than 0.05). Similarly, for NAIEs, G, T, Y and V are observed in higher abundance (Welch's t-test, p value less than 0.05).

The positional conservation of amino acids in AVPs and Non-AVPs is examined by using a two sample logo (TSL) analysis, as shown in Fig. 6. W and L are found as highly conserved at N-terminal of AVPs, whereas, F and K are found at C-terminal of AVPs. In contrast, P and N are highly conserved at N-terminal of Non-AVPs, whereas, P is conserved at C-terminal of Non-AVPs.

C. Analysis of LSTM Module

We analyze three different LSTM models: the single directional LSTM (UnidLSTM), the bi-directional LSTM (BidLSTM) and the multi-layer LSTM (MultiLSTM). The single directional LSTM model is comprised of a single hidden LSTM layer followed by a standard feed-forward output layer. The multi-layer LSTM is an extension to this model that has multiple hidden LSTM layers where each layer contains multiple memory cells. Here, we add two fully-connected layers, and use the cross-entropy as loss function to optimize defined model. Because of variable-length peptides, we put in the $L \times 20$ binary matrix for each sequence sample. The recurrent neural network decides the LSTM cell cycle-index according to sequence length.

We compare the performance of three LSTM models on the real dataset as training set $T^{544p+407n}$ and testing set $V^{60p+45n}$. During 5-fold cross validation, all three LSTM models are made on the training set $T^{544p+407n}$. As shown in Table I, UnidLSTM achieves best performance with 84.2% accuracy and 0.68 correlation; BidLSTM performs well with 83.5% accuracy and 0.66 correlation; MultiLSTM has 83.0% accuracy and 0.65 correlation. During independent evaluation, all three LSTM models are made on the testing set $V^{60p+45n}$. As shown in Table I, UnidLSTM and MultiLSTM achieve the performance with 81.9% accuracy and 0.63 correlation. However, BidLSTM performs outstanding with 85.7% accuracy and 0.71 correlation. This suggests that it may be better to choose BidLSTM as the sequence module in our method for the antiviral peptide prediction.

D. Analysis of CONV Module

We analyze four different CONV models: the dynamic CONV with original BLOSUM (DynEvo), the dynamic CONV with original PHYSICO (DynPhy), the static CONV with original BLOSUM (StaEvo), the static CONV with original PHYSICO (StaPhy). In the dynamic model, the convolutional neural network can change according to sequence length. In the static model, it can only deal with fixed-length sequence. Here, we add two fully-connected layers, and use the cross-entropy as loss function to optimize defined model. Moreover, we use evolution kernel (BLOSUM) [47] and physicochemical property kernel (PHYSICO) [50].

We compare the performance of six CONV models on the real dataset as training set $T^{544p+407n}$ and testing set $V^{60p+45n}$. During 5-fold cross validation, all four CONV models are made on the training set $T^{544p+407n}$. As shown in Table II, DynEvo and StaEvo perform well with 77.1%–77.7% accuracy and 0.53–0.55 correlation. During independent evaluation, all four CONV models are made on the testing set $V^{60p+45n}$. As shown in Table II, DynEvo and DynPhy achieve best performance

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

TABLE II
PERFORMANCE OF SIX CONV MODELS ON THE REAL DATASET

Data set	Model	SN	SP	ACC	MCC
$T^{544p+407n}$	DynEvo	81.6%	71.0%	77.1%	0.53
	DynPhy	82.0%	62.7%	73.7%	0.46
	StaEvo	81.4%	72.8%	77.7%	0.55
	StaPhy	85.1%	54.6%	72.0%	0.42
$T^{60p+45n}$	DynEvo	91.7%	73.3%	83.8%	0.67
	DynPhy	86.7%	73.3%	81.0%	0.61
	StaEvo	90.0%	66.7%	80.0%	0.59
	StaPhy	86.7%	44.4%	68.6%	0.35

TABLE III
PERFORMANCE OF TWO JOINT MODELS ON TWO BENCHMARK DATASETS

Data set	Model	SN	SP	ACC	MCC
$T^{544p+407n}$	DeepEvo	84.6%	82.1%	83.5%	0.66
	DeepPhy	85.5%	79.7%	83.0%	0.65
$T^{544p+544n*}$	DeepEvo	89.3%	90.8%	90.1%	0.80
	DeepPhy	88.0%	89.0%	88.5%	0.77
$T^{60p+45n}$	DeepEvo	90.0%	84.4%	87.6%	0.75
	DeepPhy	83.3%	75.6%	80.0%	0.59
$T^{60p+60n*}$	DeepEvo	96.7%	90.0%	93.3%	0.87
	DeepPhy	88.3%	90.0%	89.2%	0.78

with 81.0%–83.8% accuracy and 0.61–0.67 correlation. This suggests that it may be better to choose the dynamic model with BLOSUM kernel as evolution module in our method for the antiviral peptide prediction.

E. Performance of Joint Module

We analyze two different dual-channel deep neural network ensemble models: the joint model with BLOSUM (DeepEvo) and the joint model with PHYSICO (DeepPhy). Here, we produce the joint modules via training CONV channel by evolution model (BLOSUM) and physicochemical property model (PHYSICO).

We compare the performance of dual-channel model on the real dataset (training set $T^{544p+407n}$ and testing set $V^{60p+45n}$) and the random dataset (training set $T^{544p+544n*}$ and testing set $V^{60p+60n*}$). During 5-fold cross validation, two joint models are made on the training sets $T^{544p+407n}$ and $T^{544p+544n*}$. As shown in Table III, DeepEvo achieves best performance with 83.5% accuracy and 0.66 correlation on $T^{544p+407n}$, and 90.1% accuracy and 0.80 correlation on $T^{544p+544n*}$. However, DeepPhy performs well with 83.0% accuracy and 0.65 correlation on $T^{544p+407n}$, and 88.5% accuracy and 0,77 correlation on $T^{544p+544n*}$. During independent evaluation, two joint models are made on the testing sets $V^{60p+45n}$ and $V^{60p+60n*}$. As shown in Table III, DeepEvo achieves best performance with 87.6% accuracy and 0.75 correlation on $T^{60p+45n}$, and 93.3% accuracy and 0.87 correlation on $T^{60p+60n*}$. However, DeepPhy performs well with 80.0% accuracy and 0.59 correlation on $T^{60p+45n}$, and 89.2% accuracy and 0,78 correlation on $T^{60p+60n*}$. This proves once more that it may be better to choose the CONV channel with BLOSUM for building dual-channel model in our method for the antiviral peptide prediction.

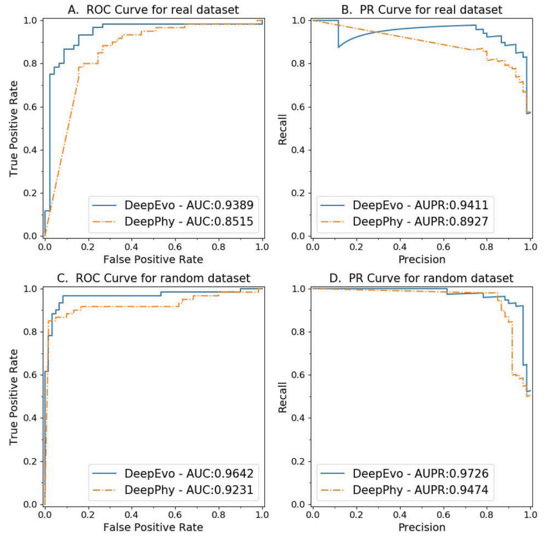

Fig. 7. The ROC and PR curves of two joint models on two benchmark datasets.

Also, we compare ROC and PR curves of dual-channel deep neural network ensemble model on two benchmark datasets, as shown in Fig. 7. On the real dataset, DeepEvo performs well with 0.9389 AUC and 0.9411 AUPR, but DeepPhy has 0.8515 AUC and 0.8927 AUPR. On the random dataset, DeepEvo achieves best performance with 0.9642 AUC and 0.9726 AUPR, but DeepPhy has 0.9231 AUC and 0.9474 AUPR. This denotes that the evolution information is much better to describe the antiviral peptides rather than physicochemical property.

F. Fine-Tuned Kernel on CONV Channel

On CONV channel, we train out fine-tuned BLOSUM and PHYSICO kernels from original matrices, such as BLOSUM* and PHYSICO*. The fine-tuned BLOSUM and PHYSICO kernels are obtained from the output of first convolutional layer of the dynamic CONV with original BLOSUM (DynEvo) and the dynamic CONV with original PHYSICO (DynPhy). Therefore, we use fine-tuned matrix to initialize first convolutional layer of the CONV module. As above, accuracy of StaEVO with fine-tuned BLOSUM matrix is better than accuracy of StaEVO with original BLOSUM matrix. Similarity, accuracy of StaPhy with fine-tuned PHYSICO matrix is better than accuracy of StaPhy with original PHYSICO matrix. It demonstrates that fine-tuned matrix is suitable to the benchmark dataset. As shown in Fig. 8, fine-tuned BLOSUM matrix differs from original BLOSUM matrix on several individual amino acid substitutional relations, but fine-tuned PHYSICO matrix differs from original PHYSICO matrix on some physicochemical properties for the vast majority of amino acids. This implies that fine-tuned PHYSICO matrix on CONV channel may have overfitting problem.

G. Comparison With Existing Predictors on the Nishant's Dataset

We compare our method, DeepAVP, with recent antiviral peptide prediction methods, such as AVPpred [7], Chang

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

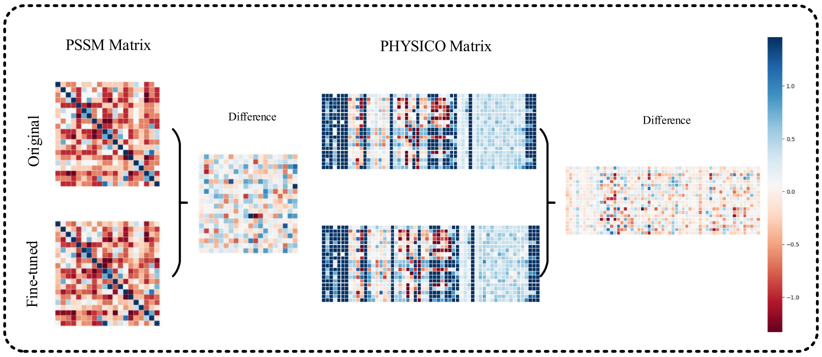

Fig. 8. Difference of original and fine-tuned matrices of BLOSUM and PHYSICO on CONV channel.

TABLE IV
COMPARISON OF OUR METHOD (DEEPAVP) WITH RECENT ANTIVIRAL PEPTIDE PREDICTION METHODS ON INDEPENDENT EVALUATION

Data set	Method	SN	SP	ACC	MCC
$T^{60p+45n}$	DeepAVP	90.0%	84.4%	87.6%	0.75
	AVPpred	88.3%	82.2%	85.7%	0.71
	Chang KY(RF)	93.3%	77.8%	86.7%	0.73
	Zare M(Adaboost)	86.2%	89.1%	87.0%	0.75
$T^{60p+60n*}$	DeepAVP	96.7%	90.0%	93.3%	0.87
	AVPpred	93.3%	91.7%	92.5%	0.85
	Chang KY(RF)	90.0%	95.0%	92.5%	0.85
	AntiVPP 1.0	87.0%	97.0%	93.0%	0.87

KY(RF) [8], Zare M(Adaboost) [9] and AntiVPP 1.0 [10]. During independent evaluation, above methods are tested on the real dataset (training set $T^{544p+407n}$ and testing set $V^{60p+45n}$) and the random dataset (training set $T^{544p+544n*}$ and testing set $V^{60p+60n*}$).

As shown in Table IV, DeepAVP achieves best performance with 87.6% accuracy and 0.75 correlation on the real dataset ($T^{60p+45n}$), and 93.3% accuracy and 0.87 correlation on the random dataset ($T^{60p+60n*}$). However, AVPpred performs well with 85.7% accuracy and 0.71 correlation on $T^{60p+45n}$, and 92.5% accuracy and 0.85 correlation on $T^{60p+60n*}$; Chang KY(RF) performs well with 86.7% accuracy and 0.73 correlation on $T^{60p+45n}$, and 92.5% accuracy and 0.85 correlation on $T^{60p+60n*}$; Zare M(Adaboost) performs well with 87.0% accuracy and 0.75 correlation on $T^{60p+45n*}$. AntiVPP performs well with 93.0% accuracy and 0.87 correlation on $T^{60p+60n*}$. This proves that dual-channel deep neural network ensemble model incorporates some unique features to outperform other methods for the antiviral peptide prediction.

H. Comparison With Existing Predictors on the Novel Dataset

We compare DeepAVP and AVPpred on the novel non-redundant AVPs dataset (413 AVPs and 348 non-AVPs); however, other three antiviral peptide prediction methods did not provide effective online service tools. As shown in Table V, DeepAVP achieves best performance with 92.4% accuracy and 0.85 correlation on the novel dataset. However, AVPpred performs well with 87.6% accuracy and 0.76 correlation on the novel dataset. This proves that dual-channel deep neural network ensemble model may have a strong competitive edge in identifying antiviral peptides on the non-redundant data.

TABLE V
COMPARISON OF DEEPAVP AND AVPPRED ON THE NOVEL DATASET

Method	SN	SP	ACC	MCC
DeepAVP	91.5%	93.4%	92.4%	0.85
AVPpred	81.1%	49.8%	87.6%	0.76

IV. CONCLUSION

We report DeepAVP, a computational approach based on dual-channel deep neural network for modeling the source of peptide antiviral variability. It constructs recurrent neural network and convolutional neural network side-by-side, recognizing all original variable-length sequence data. Applying it to the experimentally verified dataset, our highly adaptable predictor, DeepAVP, demonstrates state-of-the-art performance in identifying antiviral peptides.

Important characteristic of our model is that we processe sequence data with no need for feature extraction, whereas LSTM and CONV channels can analyze peptide sequences from sequential and evolutionary levels, respectively. Traditional method generally separates feature extraction from learning model, which leads to become a more labor-intensive task, because we cannot know in advance whether this feature extraction method is beneficial to current model. However, if we embed feature extraction into the neural network model, which can be optimized when training the learning model, which seems to have more rationality.

Furthermore, the PSSM feature extraction layer in the CONV channel can transform original BLOSUM matrix into specific evolutionary substitution matrix for antiviral peptides. We can also use this strategy to generate refined BLOSUM matrix in order to fit different peptide sequence learning task. In the future, we can use the concept of dynamic neural network to combine more feature extraction methods with various deep neural network models. It is very interesting that we will be able to perform feature extraction more effectively.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.

References

[1] R. Eléonore *et al.*, "Antiviral drug discovery strategy using combinatorial libraries of structurally constrained peptides," *J. Virology*, vol. 78, no. 14, pp. 7410–7417, 2004.

[2] C. Guillaume, C. Mohamed, H. Bernadette, and T. NoL, "Phage display of combinatorial peptide libraries: Application to antiviral research," *Molecules*, vol. 16, no. 5, pp. 3499–518, 2011.

[3] T. Nishant, Q. Abid, and K. Manoj, "VIRsiRNAdb: A curated database of experimentally validated viral siRNA/shRNA," *Nucleic Acids Res.*, vol. 40, pp. D230–D236, 2012.

[4] S. Saheli, "Vaccination: The present and the future," *Yale J. Biol. Med.*, vol. 84, no. 4, pp. 353–359, 2011.

[5] H. Jiang *et al.*, "Inhibition of influenza virus replication by constrained peptides targeting nucleoprotein," *Antiviral Chemistry Chemotherapy*, vol. 22, no. 3, pp. 119–130, 2011.

[6] Q. Abid, N. Thakur, H. Tandon, and M. Kumar, "AVPdb: A database of experimentally validated antiviral peptides targeting medically important viruses," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D1147–D1153, 2014.

[7] T. Nishant, Q. Abid, and K. Manoj, "AVPpred: Collection and prediction of highly effective antiviral peptides," *Nucleic Acids Res.*, vol. 40, pp. W199–W204, 2012.

[8] K. Y. Chang and J. R. Yang, "Analysis and prediction of highly effective antiviral peptides based on random forests," *PLOS One*, vol. 8, no. 8, 2013, Art. no. e70166.

[9] M. Zare, H. Mohabatkar, F. K. Faramarzi, M. M. Beigi, and M. Behbahani, "Using Chou's pseudo amino acid composition and machine learning method to predict the antiviral peptides," *Open Bioinformat. J.*, vol. 9, no. 1, pp. 13–19, 2015.

[10] J. F. B. Lissabet, L. H. Belén, and J. G. Farias, "Antivpp 1.0: A portable tool for prediction of antiviral peptides," *Comput. Biol. Med.*, vol. 107, pp. 127–130, 2019.

[11] J. C. Jones, E. W. Settles, C. R. Brandt, and S. C. Stacey, "Identification of the minimal active sequence of an anti-influenza virus peptide," *Antimicrobial Agents Chemotherapy.*, vol. 55, no. 4, pp. 1810–1813, 2011.

[12] W. Kerstin *et al.*, "Identification of high-affinity PBL-derived peptides with enhanced affinity to the pa protein of influenza a virus polymerase," *Antimicrobial Agents Chemotherapy*, vol. 55, no. 2, pp. 696–702, 2011.

[13] T. Narumi *et al.*, "Conjugation of cell-penetrating peptides leads to identification of anti-hiv peptides from matrix proteins," *Bioorganic Medi. Chemistry*, vol. 20, no. 4, pp. 1468–1474, 2012.

[14] F. Bai *et al.*, "Antiviral peptides targeting the west nile virus envelope protein," *J. Virology*, vol. 81, no. 4, pp. 2047–2055, 2007.

[15] S. P.-Núñez, C. J. G.-Navarro, M. G.-Delgado, J. L. Vizmanos, J. J. Lasarte, and F. B.-Cuesta, "Peptide inhibitors of hepatitis c virus ns3 protease," *Antiviral Chemistry Chemotherapy*, vol. 14, no. 5, pp. 225–233, 2003.

[16] R. Akkarawongsa, N. E. Pocaro, G. Case, A. W. Kolb, and C. R. Brandt, "Multiple peptides homologous to herpes simplex virus type 1 glycoprotein b inhibit viral infection," *Antimicrobial Agents Chemotherapy*, vol. 53, no. 3, pp. 987–996, 2009.

[17] D. Lambert *et al.*, "Peptides from conserved regions of paramyxovirus fusion (f) proteins are potent inhibitors of viral fusion," *Proc. Nat. Acad. Sci.*, vol. 93, no. 5, pp. 2186–2191, 1996.

[18] S. Popovic, E. Urbán, M. Lukic, and J. M. Conlon, "Peptides with antimicrobial and anti-inflammatory activities that have therapeutic potential for treatment of acne vulgaris," *Peptides*, vol. 34, no. 2, pp. 275–282, 2012.

[19] C. D. Fjell, J. A. Hiss, R. E. Hancock, and G. Schneider, "Designing antimicrobial peptides: Form follows function," *Nature Rev. Drug Discovery*, vol. 11, no. 1, pp. 37–51, 2012.

[20] L. A.-Mendoza *et al.*, "Overlap and diversity in antimicrobial peptide databases: Compiling a non-redundant set of sequences," *Bioinformatics*, vol. 31, no. 15, pp. 2553–2559, 2015.

[21] S. Thomas, S. Karnik, R. S. Barai, V. K. Jayaraman, and S. I.-Thomas, "Camp: A useful resource for research on antimicrobial peptides," *Nucleic Acids Res.*, vol. 38, no. suppl_1, pp. D774–D780, 2009.

[22] P. Wang *et al.*, "Prediction of antimicrobial peptides based on sequence alignment and feature selection methods," *PloS One*, vol. 6, no. 4, 2011, Art. no. e18476.

[23] W. Lin and D. Xu, "Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types," *Bioinformatics*, vol. 32, no. 24, pp. 3745–3752, 2016.

[24] M. N. Gabere and W. S. Noble, "Empirical comparison of web-based antimicrobial peptide prediction tools," *Bioinformatics*, vol. 33, no. 13, pp. 1921–1929, 2017.

[25] L. Hui *et al.*, "MeT-DB V2.0: Elucidating context-specific functions of n6-methyl-adenosine methyltranscriptome," *Nucleic Acids Res.*, no. 46, pp. D281–D287, 2018.

[26] M. S.-Castillo, D. Blanco, I. M. T.-Luna, M. C. Carrion, and Y. Huang, "A Bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data," *Bioinformatics*, vol. 34, no. 6, pp. 964–970, 2017.

[27] Y. Ding, J. Tang, and F. Guo, "Identification of drug-side effect association via semi-supervised model and multiple kernel learning," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 6, pp. 2619–2632, Nov. 2019.

[28] M. Mohammadi and A. Mansoori, "A projection neural network for identifying copy number variants," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 5, pp. 2182–2188, Sep. 2019.

[29] H. Lei *et al.*, "Protein–protein interactions prediction via multimodal deep polynomial network and regularized extreme learning machine," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 3, pp. 1290–1303, May 2019.

[30] D. Raví *et al.*, "Deep learning for health informatics," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 4–21, Jan. 2017.

[31] M. B. Pouyan and M. Nourani, "Clustering single-cell expression data using random forest graphs," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 4, pp. 1172–1181, Jul. 2017.

[32] S. K. Sønderby, C. K. Sønderby, H. Nielsen, and O. Winther, "Convolutional LSTM networks for subcellular localization of proteins," in *Proc. Int. Conf. Algorithms Computat. Biol.*, 2015, pp. 68–80.

[33] M. Spencer, J. Eickholt, and J. Cheng, "A deep learning network approach to ab initio protein secondary structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 12, no. 1, pp. 103–112, Jan./Feb. 2015.

[34] Y. Han and D. Kim, "Deep convolutional neural networks for pan-specific peptide-MHC class I binding prediction," *BMC Bioinformat.*, vol. 18, no. 1, 2017, Art. no. 585.

[35] D. Veltri, U. Kamath, and A. Shehu, "Deep learning improves antimicrobial peptide recognition," *Bioinformatics*, vol. 15, pp. 2740–2747, 2018.

[36] B. Bulik-Sullivan *et al.*, "Deep learning using tumor HLA peptide mass spectrometry datasets improves neoantigen identification," *Nature Biotechnol.*, vol. 37, no. 1, pp. 55–63, 2019.

[37] S. Lata, N. K. Mishra, and G. P. Raghava, "AntiBP2: Improved version of antibacterial peptide prediction," *BMC Bioinformat.*, vol. 11, no. Suppl 1, pp. 1–7, 2010.

[38] T. Shaini, K. Shreyas, B. R. Shankar, V. K. Jayaraman, and I. T. Susan, "CAMP: A useful resource for research on antimicrobial peptides," *Nucleic Acids Res.*, vol. 38, pp. D774–D780, 2010.

[39] W. Guangshun, L. Xia, and W. Zhe, "APD2: The updated antimicrobial peptide database and its application in peptide design," *Nucleic Acids Res.*, vol. 37, pp. 933–937, 2009.

[40] Q. A, T. N, H. T, and K. M, "AVPdb: A database of experimentally validated antiviral peptides targeting medically important viruses," *Nucl. Acids Res.*, vol. 42, pp. D1147–D1153, 2013.

[41] G. Wang, X. Li, and Z. Wang, "APD3: The antimicrobial peptide database as a tool for research and education," *Nucl. Acids Res.*, vol. 44, pp. D1087–D1093, 2016.

[42] F. H. Waghu, R. S. Barai, P. Gurung, and S. Idiculathomas, "CAMPR3: A database on sequences, structures and signatures of antimicrobial peptides," *Nucleic Acids Res.*, vol. 44, pp. D1094–D1097, 2016.

[43] X. Zhao, H. Wu, H. Lu, G. Li, and Q. Huang, "LAMP: A database linking antimicrobial peptides," *PLOS One*, vol. 8, no. 6, 2013, Art. no. e66557.

[44] W. Li and A. Godzik, "Cd-Hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht, "Use of the 'perceptron' algorithm to distinguish translational initiation sites in E. Coli," *Nucleic Acids Res.*, vol. 10, no. 9, pp. 2997–3011, 1982.

[47] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc. Nat. Acad. Sci. USA*, vol. 89, no. 22, pp. 10 915–10 919, 1992.

[48] Y. Shen, J. Tang, and F. Guo, "Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC," *J. Theor. Biol.*, vol. 462, pp. 230–239, 2019.

[49] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[50] E. Gasteiger *et al.*, *Protein Identification and Analysis Tools on the ExPASy Server*, J. M. Walker, Ed., Totowa, NJ, USA: Humana Press, 2005.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 20,2022 at 19:34:45 UTC from IEEE Xplore. Restrictions apply.