Chapter 2

Accelerating the Discovery of Peptide-based Antifungal Drugs using Artificial Intelligence

Fungal infections cause a wide range of diseases in humans and animals. Following the onset of the COVID-19 pandemic in 2019, the rates of fungal infections have increased significantly. Many people who recovered from COVID-19 disease got infected by black, white, or yellow fungal infections, and nearly 50,000 black fungus infection cases have been reported in India alone. In many cases of fungal infection, the patients do not respond to available antifungal drugs due to antifungal drug resistance. Because of antifungal resistance, antifungal peptides (AFPs) have recently received much interest as an alternative to existing antifungal drugs. A diverse population of living organisms produces AFPs. Wet lab researchers conduct various trials in the lab to identify novel AFPs 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 AFPs, the in-silico tool is needed. Thus, this chapter introduces a deep-learning model named Deep-AFPpred, which has been made available as a web

app at https://afppred.anvil.app/. This app will aid wet lab researchers in the fight against antifungal resistance by accelerating the discovery of peptide-based antifungal medications.

2.1 Introduction

Fungi are ubiquitous eukaryotic organisms that form an integral part of our commensal microbiota. These can be found in many parts of the body, such as the lungs, stomach, skin, oral cavity, and reproductive tract. These have been employed for thousands of years for food processing [28] and in a variety of industrial activities, including the production of peptides, antibiotics, organic acids, enzymes, and vitamins [29, 30].

However, fungal infections or mycosis cause a wide range of diseases in humans and animals, ranging from superficial and subcutaneous infections to life-threatening systemic infections [31]. The situation becomes more intense when there is mycosis in immunocompromised patients suffering from autoimmune diseases. Following the onset of the COVID-19 pandemic in 2019, the rates of community-acquired and nosocomial fungal infections have increased significantly. In COVID-19 conditions, a person's immune system gets weakened, which prevents the body from effectively protecting against infections. As a result, black, white, and yellow fungal infection is reported in many people recovering from COVID-19 disease. In India alone, nearly fifty thousand black fungus infection (mucormycosis) cases have been reported to date. Moreover, some states in the country have declared fungal infection as a notifiable disease under the Epidemic Diseases Act 1987.

Among fungal pathogens that infect humans, Candida albicans, Candida auris, Cryptococcus neoformans, and Aspergillus fumigatus are the major ones, while other agents like Malassezia furfur and Histoplasma capsulatum are also emerging [32]. Millions of death have been reported due to fungal infections each year [33, 34] and four major classes of antifungal agents: azoles, polyenes, echinocandins, and fluorinated

pyrimidines are available in the market to treat such infections [32]. In many cases of fungal infection, the patients do not respond to available antifungal drugs due to antifungal drug resistance in the causative organisms. Antifungal drug resistance occurs due to the overuse of available antifungal drugs. To secure their survival in the presence of antifungal medicines, the fungus undergoes various changes in their genome. As a result, the antifungal drugs are no longer effective against the fungi. Because of the issue of antifungal resistance, AFPs have received a lot of interest recently as an alternative to existing antifungal drugs.

AFPs are produced by a diverse population of living organisms. Wet lab researchers conduct various trials in the lab to identify novel AFPs from these natural resources, which involves a lot of time and money. As a result, to undertake a preliminary screening of natural sources with the purpose of discovering potential AFPs, the *in-silico* tool is needed. In literature, tools namely iAMPpred [17], Antifp_Main [18], Antifp_DS2 [18], and PhytoAFP [19] 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. (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 AFPs has expanded significantly. However, the existing tools have utilized only a few of the available AFPs present in the literature. (iii) Only a few keywords were considered while developing a filter for extracting the Non-AFPs from the literature, which may have caused even AFPs to cross the filter.

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 their time and money. 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 proposed a framework (the model obtained from the proposed framework is termed as Deep-AFPpred) that utilizes the concept of transfer learning with the 1DCNN-BiLSTM deep learning algorithm. The concept of transfer learning was accomplished by utilizing pretrained embeddings from seq2vec (PESTV). Authors in [14] learned PESTV by training Embeddings from Language Models (ELMo) on millions of protein sequences from UniRef50. To understand the contribution of transfer learning in the proposed framework, we performed the ablation studies by utilizing non-pretrained encodings from PAM250, BLOSUM62, and one-hot encoding (OHE) with a 1DCNN-BiLSTM deep learning algorithm. We found that performance decreases when the transfer learning technique is eliminated. Additionally, we also experimented with various machine-learning algorithms (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 pro-

posed framework performed better than all others.

The significant contributions of this chapter are as follows: 1) We proposed a model named Deep-AFPpred, 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 hand-crafted features (HCF) for making predictions, thus removing our reliance on domain expertise. 3) We conducted ablation studies, experimented with various machine-learning algorithms, and found that our proposed framework performed better. 4) We also discovered novel AFPs by screening ten antifungal proteins from five distinct genera (two antifungal proteins from each genus). Based on specific selection criteria, we also recommend one AFP from each protein for wet lab synthesis and evaluation of the antifungal activity. 5) To assist wet-lab researchers in identifying novel AFPs from any protein, the proposed model Deep-AFPpred has also been made available as a web server at https://afppred.anvil.app/. This web server is made flexible, wherein users can customize various parameters to identify AFPs from protein sequences.

The rest of this chapter is arranged as follows. Section 2.2 provides details of the dataset, peptide encoding, and proposed framework. The details of experimental configuration, performance metrics, assessment procedure, results obtained from the proposed framework, results obtained from the ablation studies, results obtained from the additional experiments, generalization performance of our proposed model along with existing AFP prediction tools on test data are presented in Section 2.3. The identification of AFPs in the antifungal proteins utilizing our proposed model is presented in Section 2.4. The details about the web server are provided in Section 2.5. The conclusion is provided in Section 2.6.

2.2 Materials and Methods

2.2.1 Dataset Collection

In this study, we collected 3441 unique AFPs of length ∈ [5,30] amino acids containing natural amino acids from CAMP [41], DRAMP [42] and StarPep [43, 44] databases. We obtained 3441 unique non-AFPs of length ∈ [5,30] containing natural amino acids from the Swiss-Prot database [45]. These non-AFPs were collected from the reviewed, manually annotated proteins from the Swiss-Prot that did not contain any of the following keywords: antifungal, antimicrobial, antibacterial, antiviral, antibiotic, antitoxin, antitumor, defensin, anti-TB, anti-HIV, antimalarial, anticancer, antiendotoxin, anti-diabetic, insecticidal, cytokine, antioxidant, anti-MRSA, anti-gram +, anti-gram-, anti-protist, antiprotozoal, bacteriocin, antibiofilm, anti-inflammatory, antiparasitic, secreted, excreted, effector.

Thus our dataset contained 6882 peptides (AFPs: 3441, non-AFPs: 3441), which was further divided into two sets, namely Training set (S^{Train}) and Test set (S^{Test}). S^{Train} contains 60% peptides and can be defined as follows:

$$
\begin{aligned}
& S^{Train} = S^{Train}_{AFPs} \cup S^{Train}_{non-AFPs} \\
& \text{where,} \\
& S^{Train}_{AFPs} \cap S^{Train}_{non-AFPs} = \emptyset \\
& |S^{Train}_{AFPs}| = 2062 \\
& |S^{Train}_{non-AFPs}| = 2062 \\
& |S^{Train}| = 4124
\end{aligned}
\tag{2.1}
$$

S^{Test} contains remaining 40% peptides and can be defined as follows:

$$
\begin{aligned}
&S^{Test} = S^{Test}_{AFPs} \cup S^{Test}_{non-AFPs} \\
&\text{where,} \\
&S^{Test}_{AFPs} \cap S^{Test}_{non-AFPs} = \emptyset \\
&|S^{Test}_{AFPs}| = 1379 \\
&|S^{Test}_{non-AFPs}| = 1379 \\
&|S^{Test}| = 2758
\end{aligned}
\tag{2.2}
$$

Moreover, there is no overlap between S^{Train} and S^{Test}, as shown in Equation 2.3.

$$
S^{Train} \cap S^{Test} = \emptyset \tag{2.3}
$$

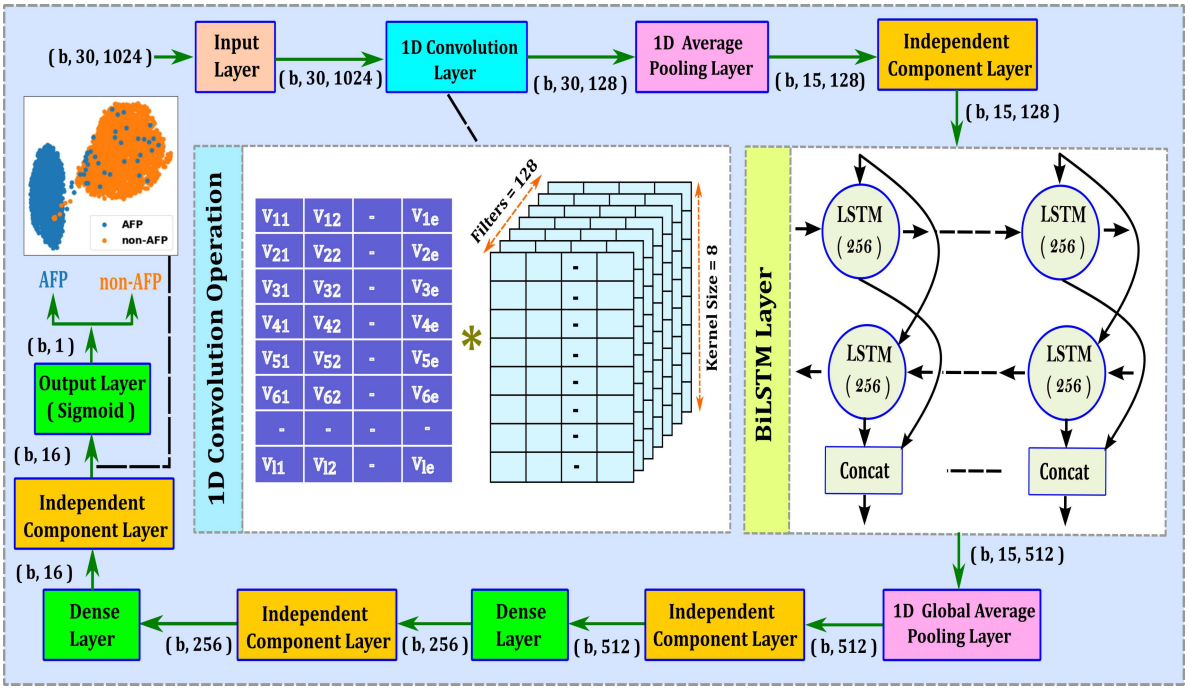

Figure 2.1: Proposed Framework

2.2.2 Proposed Framework

The proposed architecture for Deep-AFPpred is shown in Figure 2.1, in which the output of one layer is provided as input to the subsequent layer. The first layer is the

Input Layer through which we fed the data (PESTV, each of length 1024 in our case). The data provided via this layer must have the same dimension. Therefore we need to create PESTV from all peptides of the identical dimension. We have considered the peptides of length $\in$ [5,30]; therefore, the maximum value for the peptide length is 30. Therefore to make PESTV from all peptides of uniform dimension, we have performed post padding with zero vector(s) each of length 1024 (because PESTV has length 1024) to the PESTV from peptides whose length is less than 30. As a result, this layer outputs a three-dimensional tensor of shape (b, 30, 1024), as shown in Figure 2.1, where b denotes the batch size (= 16 identified via hyperparameter tuning).

The second layer is 1D Convolution Layer. The job of this layer is to recognize the various patterns in the peptides. The two essential components in this layer are (i) Filter size: signifies the number of amino acids to be considered at once while identifying patterns. (ii) Number of filters: one filter is insufficient to identify all of the patterns contained in the peptides; therefore, several filters are required. We have treated both filter size and the number of filters as hyperparameters and obtained the optimal values as 8 and 128, respectively. This layer involves an essential operation known as the 1D convolution operation, which takes place between the PESTV of peptide(s) and different filters. The weights of filters were randomly initialized, which get updated as the training progresses. After the convolution operation rectified linear unit (ReLU) activation function is used.

While performing convolution operation, the value of padding and strides is taken as "same" and 1, respectively; therefore, corresponding to one convolution filter, we obtained a vector of size 30. We have used 128 convolution filters, so we obtained 128 vectors, each of size 30. As a result, this layer outputs a three-dimensional tensor of shape (b, 30, 128), as shown in Figure 2.1.

The third layer is 1D Average Pooling Layer. This layer is used with the filter of size 2 and strides = 2, which down-samples the input representation by sliding a

non-overlapping window of length 2 and taking the average of the values present in the window. As a result, this layer outputs a three-dimensional tensor of shape (b, 15, 128), as shown in Figure 2.1.

The fourth layer is Independent Component Layer (ICL). 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 a more stable training process, faster convergence speed, and better generalization performance. Later several people have also used this concept and found it effective [47]. In our work, we have also practiced the ICL with a dropout rate of 0.25 (identified via hyperparameter tuning) and found improvement in the model's performance. This layer makes no modifications to the shape of data provided to it; hence, the shape of output from this layer is the same as the shape of input to this layer, i.e., (b, 15, 128), as shown in Figure 2.1.

The fifth layer is BiLSTM Layer. This layer comprises three sublayers: forward layer, backward layer, and concatenation layer. Both forward and backward layers are made up of a chain of 15 repeating units (also known as cells), and the concatenation layer combines their output. At each time step t ($1 \leq t \leq 15$) x_t, h_{t-1}, and c_{t-1} are the inputs to cell and h_t, c_t are the outputs. Where x_t is the 128-dimensional input vector to the cell obtained from ICL. The hidden state and cell state at time steps t-1 and t, respectively, are represented by (h_{t-1}, c_{t-1}) and (h_t, c_t). The calculation of h_t and c_t is provided in the Section 1.2.2.2. The number of neurons in both the forward and backward layer each are 256 (identified via hyperparameter tuning) followed by the concatenation layer; therefore, this layer outputs a three-dimensional tensor of shape (b, 15, 2*256), as shown in Figure 2.1.

The sixth layer is 1D Global Average Pooling Layer. This layer is used to accomplish downsampling by obtaining the average value over the time dimension. As illustrated

in Figure 2.1, this layer produces a two-dimensional tensor of shape (b, 512).

Next, ICL with a dropout rate of 0.2 (identified via hyperparameter tuning) followed by a Dense layer comprising 256 neurons (identified via hyperparameter tuning) with ReLU activation function is applied. From here, we got a two-dimensional tensor of shape (b, 256) as output, as shown in Figure 2.1.

After that, ICL with a dropout rate of 0.15 (identified via hyperparameter tuning) followed by a Dense layer comprising 16 neurons (identified via hyperparameter tuning) with ReLU activation function is applied. From here, we got a two-dimensional tensor of shape (b, 16) as output, as shown in Figure 2.1.

Finally, ICL with a dropout rate of 0.05 (identified via hyperparameter tuning) followed by a Dense layer comprising a single neuron with 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-AFP class; otherwise, it belongs to the AFP class. As a result, we obtained a two-dimensional tensor of shape (b, 1), as shown in Figure 2.1.

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

2.3 Experiments and Results

This section briefly describes the experimental configuration, performance metrics, assessment procedure, and results obtained from the proposed framework. To understand the contribution of transfer learning in the proposed framework, we performed the ablation studies by utilizing non-pretrained encodings from PAM250, BLOSUM62, and one-hot encoding (OHE) with a 1DCNN-BiLSTM deep learning algorithm. Moreover, we have also experimented with various machine-learning algorithms. This section also provides the results from these ablation studies, machine-learning algorithms 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.

2.3.1 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.

2.3.2 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.

2.3.3 Assessment Procedure

The complete dataset consisting of 6882 peptides is divided into two sets S^{Train} and S^{Test}. S^{Train} contains 60% peptides (4124 peptides), which was used for hyperparameter tuning and identifying the best framework (1DCNN-BiLSTM + PESTV) among the frameworks available from different methods, whereas S^{Test} contains the remaining 40% peptides (2758 peptides), was used to test the generalization performance of our proposed model (obtained from the best framework) Deep-AFPpred . To use the entire data available in S^{Train} as validation data (for hyperparameter tuning and identifying the best framework), we further divided S^{Train} into five folds: Fold 1, ..., Fold 5. Using the aforementioned five-folds, we created five splits: Split 1, ..., Split 5. In each Split k ($1 \leq k \leq 5$), validation data was obtained from Fold k, and training data was obtained from the remaining four folds, which ensures that each fold is used exactly once for validation. The hyperparameters were identified on Split 1, and the same hyperparameters were utilized in the remaining splits (Split 2,..., Split 5). This

whole process provided us with five models: Model 1,..., Model 5 trained and validated utilizing Split 1, ..., Split 5, respectively. Let P1, ..., and P5 denote the performance metrics obtained from Model 1, ..., Model 5, respectively, then Mean (P_μ) ± Standard deviation (P_σ) is reported as validation performance (P_{re}), which is employed during the model selection phase to select the best model among the available models. P_{re} can be calculated as follows:

$$\begin{aligned} P_\mu &= \frac{\sum_{i=1}^{i=5} P_i}{5} \\ P_\sigma &= \sqrt{\frac{\sum_{i=1}^{i=5} (P_i - P_\mu)^2}{5}} \\ P_{re} &= P_\mu \pm P_\sigma \end{aligned} \tag{2.4}$$

2.3.4 Results obtained from proposed Framework

We proposed a framework that utilizes the concept of transfer learning in terms of PESTV with a 1DCNN-BiLSTM deep learning algorithm. The results obtained on validation data by the proposed framework are provided in Table 2.1. 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 ≈ 94, 93, 96, 94, 96, 98, and 89, respectively. Before arriving at the proposed framework, we performed ablation studies and conducted additional experiments considering state-of-the-art methods. These ablation studies and additional experiments are evaluated on the validation data, and their findings are presented in the subsequent Sections.

Table 2.1: Results obtained by utilizing PESTV with a 1DCNN-BiLSTM deep learning algorithm.

S. No.	Embedding	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	PESTV	94.27 ± 0.46	92.72 ± 0.94	95.69 ± 0.34	94.18 ± 0.50	95.82 ± 0.35	97.84 ± 0.53	88.60 ± 0.91

Table 2.2: Results obtained by utilizing amino acid encodings from PAM250, BLOSUM62, and OHE with a 1DCNN-BiLSTM deep learning algorithm.

S. No.	Encoding	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	PAM250	92.21 ± 0.60	89.91 ± 0.64	94.26 ± 0.95	92.03 ± 0.61	94.52 ± 0.94	96.89 ± 0.28	84.52 ± 1.24
2	BLOSUM62	91.24 ± 0.53	88.26 ± 0.96	93.87 ± 0.84	90.97 ± 0.56	94.22 ± 0.85	96.62 ± 0.35	82.64 ± 1.07
3	OHE	91.12 ± 0.45	87.68 ± 0.68	94.17 ± 0.93	90.80 ± 0.45	94.56 ± 0.93	96.48 ± 0.20	82.45 ± 0.94

2.3.5 Ablation Studies

The concept of transfer learning is utilized in the proposed framework. Therefore, before finalizing the proposed framework to understand the role of transfer learning in it, we performed the ablation studies using non-pretrained encodings from PAM250, BLOSUM62, and one-hot encoding (OHE) with a 1DCNN-BiLSTM deep learning algorithm. In OHE, each amino acid is represented by a vector that contains zeros in all cells except one, which uniquely identifies that specific amino acid, whereas PAM250 and BLOSUM62 are amino acid substitution matrices [50]. The results obtained on employing amino acid encodings from PAM250, BLOSUM62, and OHE with a 1DCNN-BiLSTM deep learning algorithm, are shown in Table 2.2 and the results obtained on applying PESTV with 1DCNN-BiLSTM (Proposed) is shown in Table 2.1. The comparison between them in terms of composite metrics is shown in Figure 2.2. As can be seen from Figure 2.2, PESTV + 1DCNN-BiLSTM outperforms others (PAM250 + 1DCNN-BiLSTM, BLOSUM62 + 1DCNN-BiLSTM and OHE + 1DCNN-BiLSTM). Specifically, MCC, which is the most reliable metric for evaluating binary classification task is atlest 4 % more for our proposed framework than others.

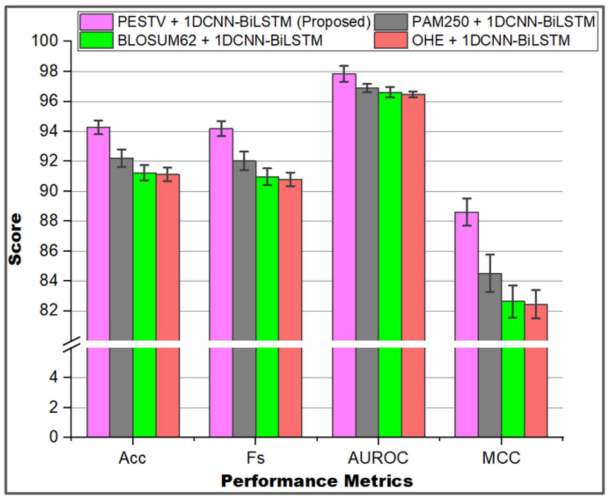

Figure 2.2: Comparison of results obtained from the 1DCNN-BiLSTM using pre-trained embeddings (PESTV) and amino acid encodings (PAM250, BLOSUM62, OHE).

2.3.6 Additional Experiments

We conducted additional experiments by considering machine learning algorithms before finalizing the proposed framework. Machine learning algorithms cannot perform automatic feature extraction, and therefore we need to provide features to machine learning algorithms explicitly known as hand crafted features (HCF). Existing AFP prediction tools have utilized the SVM algorithm with different HCF (iAMPpred utilized structural, physicochemical, and compositional properties based features with SVM. Antifp utilized N15C15 binary profile-based features with SVM. PhytoAFP utilized tripeptide composition-based features with SVM). We also constructed HCF (known as HCF1) by considering the structural, physicochemical, and compositional properties of peptides, which include alpha-helix propensity (AHP), beta-turn propensity (BTP), beta-sheet propensity (BSP), charge (CH), hydrophobicity index (HI), isoelectric point (IP), molecular weight (MW), amino acid composition (AAC), pseudo amino acid composition (PAAC) and amphiphilic pseudo amino acid composition (APAAC). Additionally, we constructed two other sets of HCF known as HCF2 and HCF3. HCF2 includes the N15C15 binary profile-based feature of peptides, and HCF3 includes the tripeptide

composition-based features of peptides. These HCF (HCF1, HCF2, and HCF3) were utilized with six diverse, widely used machine learning techniques, namely XGB, SVM, RF, LR, NB, and KNN. The results obtained from these machine learning algorithms using HCF1, HCF2, and HCF3 is shown in Table 2.3, Table 2.4, and Table 2.5. The results obtained from the proposed framework (PESTV with 1DCNN-BiLSTM) are shown in Table 2.1. The comparison between results obtained from machine learning algorithms using HCF1, HCF2, and HCF3 and proposed framework in terms of composite metrics is provided in Figure 2.3, Figure 2.4, and Figure 2.5. As can be seen from these figures, our proposed framework outperforms machine learning algorithms. Specifically, MCC, which is the most reliable metric for evaluating binary classification task is atlest 5 % more for our proposed framework than others.

Table 2.3: Results obtained by utilizing HCF1 with various machine learning algorithms.

S. No.	Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	XGB	89.91 ± 0.27	88.55 ± 0.78	91.03 ± 0.83	89.77 ± 0.26	91.27 ± 0.94	96.11 ± 0.36	79.86 ± 0.56
2	SVM	89.54 ± 0.45	89.13 ± 0.99	89.88 ± 0.64	89.50 ± 0.48	89.96 ± 0.75	95.42 ± 0.45	79.11 ± 0.90
3	RF	89.50 ± 0.74	86.66 ± 0.93	91.90 ± 1.46	89.19 ± 0.70	92.33 ± 1.55	95.96 ± 0.41	79.14 ± 1.51
4	LR	85.03 ± 0.68	85.25 ± 1.49	84.90 ± 0.88	85.06 ± 0.73	84.82 ± 1.14	91.91 ± 0.31	70.09 ± 1.38
5	NB	81.93 ± 0.53	83.99 ± 1.71	80.69 ± 1.05	82.29 ± 0.61	79.87 ± 1.63	89.90 ± 0.48	63.95 ± 1.09
6	KNN	87.65 ± 0.64	88.60 ± 0.71	86.95 ± 0.59	87.77 ± 0.64	86.71 ± 0.61	95.09 ± 0.58	75.32 ± 1.29

Table 2.4: Results obtained by utilizing HCF2 with various machine learning algorithms.

S. No.	Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	XGB	89.91 ± 1.15	88.79 ± 1.00	90.83 ± 1.54	89.80 ± 1.12	91.02 ± 1.61	95.56 ± 0.55	79.85 ± 2.31
2	SVM	90.54 ± 0.67	88.55 ± 0.95	92.23 ± 1.05	90.35 ± 0.67	92.53 ± 1.10	96.21 ± 0.46	81.16 ± 1.35
3	RF	90.66 ± 0.40	87.39 ± 1.34	93.52 ± 0.69	90.34 ± 0.50	93.93 ± 0.76	96.42 ± 0.33	81.51 ± 0.72
4	LR	86.20 ± 0.67	83.85 ± 1.66	88.00 ± 0.92	85.86 ± 0.78	88.55 ± 1.10	93.01 ± 0.76	72.50 ± 1.31
5	NB	85.91 ± 0.63	82.78 ± 2.08	88.32 ± 0.58	85.44 ± 0.86	89.04 ± 0.88	92.71 ± 0.54	71.99 ± 1.16
6	KNN	79.04 ± 2.10	92.29 ± 1.48	73.08 ± 2.65	81.52 ± 1.43	65.81 ± 4.92	91.19 ± 0.74	60.32 ± 3.52

Table 2.5: Results obtained by utilizing HCF3 with various machine learning algorithms.

S. No.	Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	XGB	86.78 ± 0.71	83.75 ± 1.48	89.19 ± 1.40	86.36 ± 0.76	89.81 ± 1.52	93.51 ± 0.36	73.73 ± 1.45
2	SVM	91.53 ± 0.17	86.80 ± 0.45	95.88 ± 0.48	91.11 ± 0.18	96.26 ± 0.47	96.94 ± 0.14	83.45 ± 0.36
3	RF	90.27 ± 0.95	88.55 ± 1.33	91.72 ± 1.21	90.10 ± 0.96	91.99 ± 1.27	96.43 ± 0.39	80.61 ± 1.89
4	LR	91.17 ± 0.08	88.26 ± 1.01	93.73 ± 0.96	90.90 ± 0.12	94.08 ± 1.05	96.40 ± 0.32	82.50 ± 0.22
5	NB	85.93 ± 0.96	90.78 ± 0.73	82.77 ± 1.25	86.59 ± 0.85	81.08 ± 1.60	95.23 ± 0.48	72.22 ± 1.87
6	KNN	84.62 ± 1.20	88.89 ± 0.93	81.94 ± 1.82	85.26 ± 1.03	80.36 ± 2.37	92.98 ± 0.35	69.53 ± 2.32

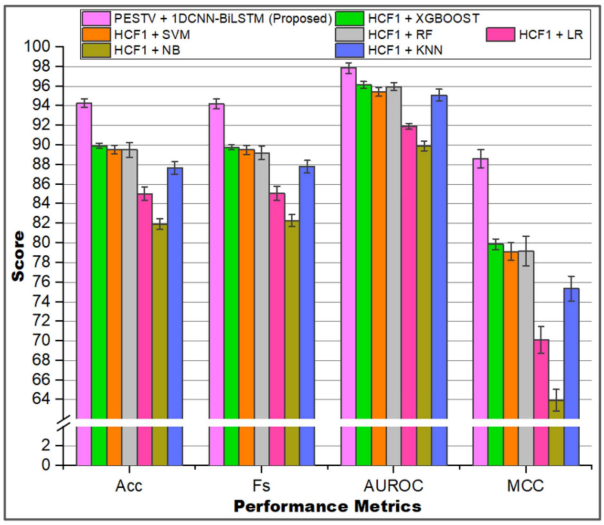

Figure 2.3: Comparison of results obtained from the PESTV + 1DCNN-BiLSTM deep learning algorithm and HCF1 + machine learning algorithms (XGBOOST, RF, NB, SVM, LR, KNN).

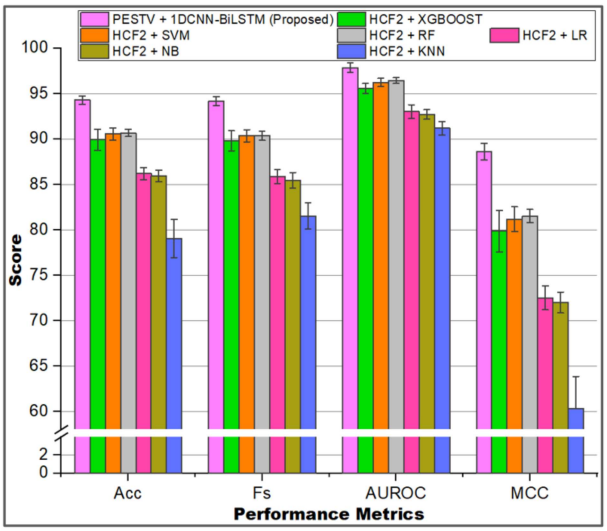

Figure 2.4: Comparison of results obtained from the PESTV + 1DCNN-BiLSTM deep learning algorithm and HCF2 + machine learning algorithms (XGBOOST, RF, NB, SVM, LR, KNN).

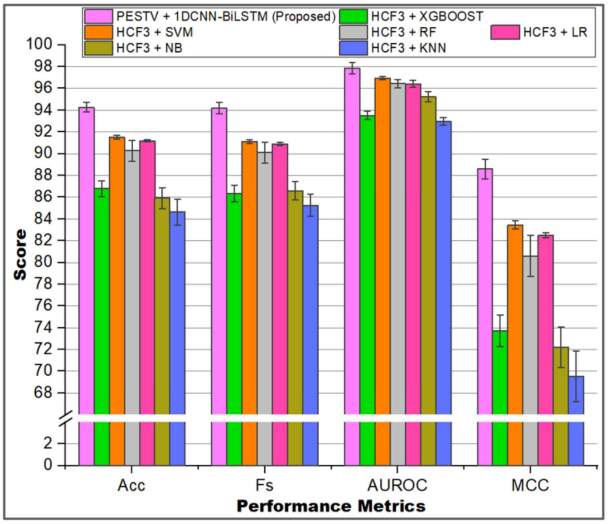

Figure 2.5: Comparison of results obtained from the PESTV + 1DCNN-BiLSTM deep learning algorithm and HCF3 + machine learning algorithms (XGBOOST, RF, NB, SVM, LR, KNN).

Table 2.6: Results obtained from proposed model Deep-AFPpred and other AFP prediction tools

S.No.	Model	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)
1.	iAMPpred	79.84	77.45	81.34	79.35	82.23	88.53
2.	Antifp_Main	51.45	22.99	53.37	32.13	79.91	48.15
3.	Antifp_DS2	56.49	32.41	62.52	42.69	80.57	59.58
4.	PhytoAFP	55.00	65.48	54.14	59.27	44.53	58.57
5.	Deep-AFPpred (Proposed)	**92.68**	**91.59**	**93.62**	**92.6**	**93.76**	**96.87**

2.3.7 Performance of proposed model Deep-AFPpred and other AFP prediction tools on test data

By performing different experiments, we found that the proposed framework PESTV + 1DCNN-BiLSTM is better than the alternatives. We have five trained models (Model 1,..., Model 5) from the proposed framework. We chose the most stable out of them to be the final model. Stability was estimated by retraining each Model 1,..., Model 5 for four more times and determining the standard deviation of the scores acquired on validation data (Fold 1,..., Fold 5 act as validation data for Model 1,..., Model 5, respectively) across five runs. The model with the lowest standard deviation (Model 2) was finally selected. This final model is termed as Deep-AFPpred and is made available to the research community as a web server. Further, we evaluated the performance of our proposed model, Deep-AFPpred, along with existing AFP prediction tools on S^{test}. As can be seen from Table 2.6, our proposed model Deep-AFPpred performed better than other AFP prediction tools. The second-best performer is iAMPpred, which achieves A_{cc}, S_n, P_r, F_s, S_p and AUROC values nearly 13, 14, 12, 13, 12 and 8% lower than our proposed model Deep-AFPpred. This better generalization performance shows that wet-lab researchers can utilize our proposed model for identifying novel AFPs from natural sources.

Table 2.7: Proposed peptides for wet-lab synthesis and experimentation.

Antifungal Protein	Sequence	Length	MW	Charge	Motif_3	Motif_4	Score_3	Score_4
Aspergillus spathulatus (SMQ11440.1) Genus: Aspergillus	YLAKCPSAANTKCEK	15	1625.92	2.8	[LAK, AAN]	-	159	-
Aspergillus niger (TPR07744.1) Genus: Aspergillus	RRTVAGGQRRPISSILGRHL	20	2229.6	6.1	[ILG, RTV, VAG, TVA, LGR]	-	317	-
Penicillium chrysogenum (KZN94175.1) Genus: Penicillium	GVLAKYTGKCTKSKNECKYK	20	2248.68	5.8	[TKS, LAK, GVL]	-	197	-
Penicillium expansum (XP_016599536.1) Genus: Penicillium	KCPSADNKKCKTDRHHCEY	19	2262.56	2.9	[KKC, KCK, RHH]	-	231	-
Colletotrichum fructicola (KAF4432206.1) Genus: Colletotrichum	GKCTRGRNYKEDTCK	15	1758	3.8	[CTR]	-	51	-
Colletotrichum spinosum (TDZ29110.1) Genus: Colletotrichum	KCTRKTNECNFTASRKKKSV	20	2328.73	6.9	[CTR, KSV, KKS, RKK]	[KKSV]	176	13
Fusarium asiaticum (CAR79023.1) Genus: Fusarium	KKCTKDGNSCKWDSASKAL	19	2069.38	3.9	[KKC, SKA, KAL, ASK]	-	291	-
Fusarium bulbicola (KAF5965014.1) Genus: Fusarium	RTAFKKCGTFANQKCTKDGA	20	2174.52	4.9	[KKC, AFK, FKK]	[FKKC]	289	16
Pyrenophora tritici-repentis (PZC87981.1) Genus: Pyrenophora	RLWSLFPELLGPESDH	16	1895.13	-0.9	[LGP, LLG]	-	175	-
Pyrenophora teres f. maculata (CAA9963916.1) Genus: Pyrenophora	VRLKFTGTCTKSTDQC	16	1787.08	2.8	[GTC, TKS]	-	86	-

2.4 Prediction of AFPs in the Antifungal Proteins

By utilizing Deep-AFPpred, we identified AFPs from the ten antifungal proteins belonging to fungi of five different genera (two antifungal proteins from one genus). The identified AFPs are listed in Table 2.7. The helical wheel representations of the proposed AFPs are provided in Figure 2.6. 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 antifungal activity of the peptide. As seen in the Figure, almost all the peptides possess hydrophobic moment. However, the peptides (SMQ11440.1, PZC87981.1, CAA9963916.1) display a good amount of hydrophobic moment and, therefore, can be considered for wet lab synthesis and experimentation.

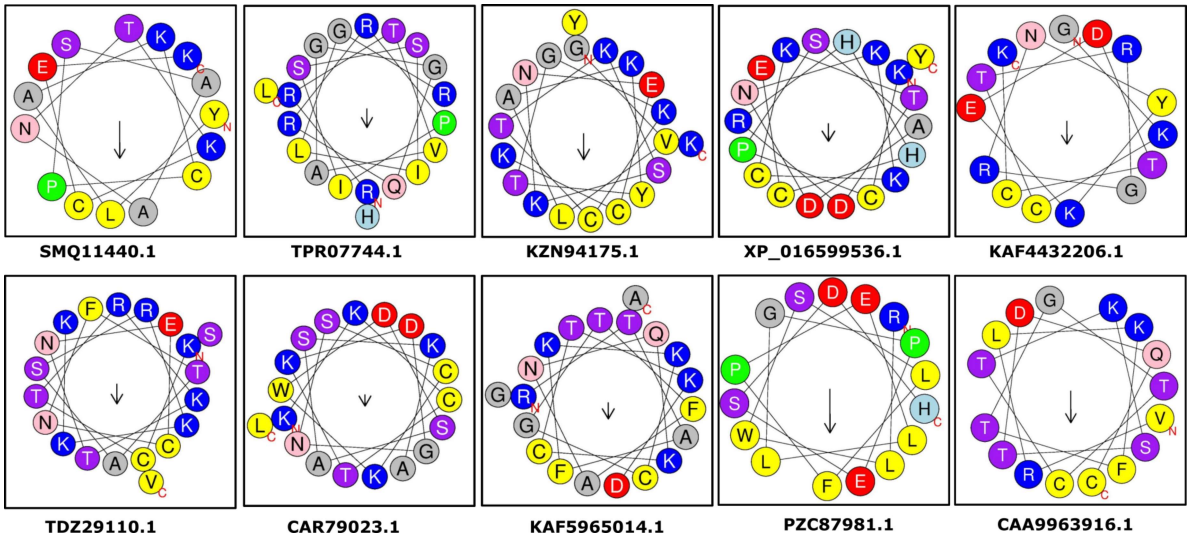

Figure 2.6: Helical wheel representation of proposed AFPs.

We obtained these antifungal proteins from the protein database of NCBI [51] and performed the following steps for identifying AFPs:

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:** The peptides from the peptide library were provided as input to Deep-AFPpred, which provided us probability value for each peptide. Only the peptides for which Deep-AFPpred provided the probability ≥ 0.99 were considered, and others were ignored.
3. **Motif Search:** We used the MERCI (Motif - EmeRging and with Classes Identification) [52] programme to find the motifs. Using MERCI, we identified 405 motifs of length 4 which are solely present in at least 10 AFP sequences. These 405 motifs cover 2208 out of 3441 AFPs, accounting for around 65 percent of all AFPs. By extracting the substrings of length 3 from these 405, 4-length motifs, we produced 479 distinct 3-length motifs. These 3-length motifs cover additional 988 AFP sequences, which were not covered by 4-length motifs. Thus, 3 and 4-length motifs cover 3196 AFPs, accounting for around 93 percent of all AFPs. We have searched for these preidentified motifs of lengths 3 and 4 amino acids in the peptides obtained from the previous step. The Score_3 and Score_4 (one corresponding to the motifs of length 3 and the other corresponding to the motifs of length 4) were obtained for each peptide based on occurrences of motifs of lengths 3 and 4, respectively, in the 3441 AFPs considered. For example, let peptide P contain two motifs, M1 and M2, of length 3, then Score_3 can be obtained by adding the occurrence of M1 and M2 in the 3441 AFPs considered (i.e., the number of AFPs among 3441 AFPs which contains M1 + number of AFPs among 3441 AFPs which contains M2) and in a similar fashion, Score_4 can also be obtained. We ignored the sequences for which both Score_3 and Score_4 come out to be zero.
4. **Solubility:** The sequences which we obtained from the previous step were checked for water solubility utilizing PepCalc (https://pepcalc.com/), and the sequences

that show poor solubility were neglected.

5. **Sorting:** Lastly, we sort the peptides obtained from the previous step in the order of decreasing Score_4 and Score_3 values (If Score_4 was the same for two peptides, then they were sorted according to Score_3) and proposed one peptide from each protein that was present at the topmost position for wet-lab synthesis and experimentation.

2.5 Web Server

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

- **Classify Peptides:** This module helps in determining whether the query peptide is AFP or Non-AFP. As shown in Figure 2.7, this module takes query peptides in raw format as input and provides Prediction Probability $\in$ [0,1] and prediction 0 (Non-AFP) or 1 (AFP) as output. If the Prediction Probability $\in$ [0,0.5], then the prediction will be returned as 0; else, 1 is returned.

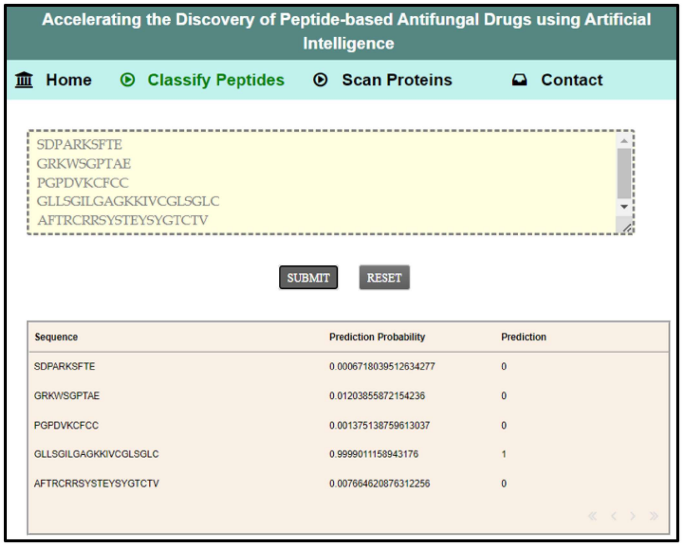

Figure 2.7: Classify query peptide as AFP/Non-AFP.

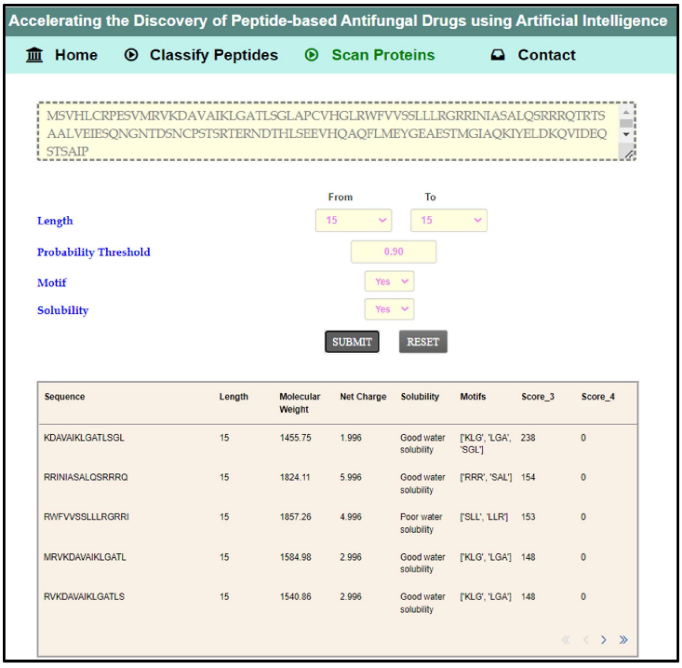

Figure 2.8: Identify AFPs from protein sequence.

- **Scan Proteins:** This module helps in identifying the novel AFPs from any protein. This module is made flexible, wherein users can customize various parameters as per requirement (as shown in Figure 2.8). These parameters include: (i) Length: Length considered while preparing peptide library. This can be length $\in$ [5, 30] (Default: length $\in$ [10, 20]). (ii) Probability: Threshold value for probability. Any value of probability > 0.5. (Default : 0.99). (iii) Motif: Yes/No, which means whether or not to perform motif search and find Score_3 and Score_4 values. (Default: Yes). (iv) Solubility: Yes/No, which means whether or not to consider water solubility (Default: Yes). Once the user enters the protein sequence along with customized parameters, the web server will generate the report, which contains the following fields: (i) Sequence: peptide sequence (ii) Length: length of peptide (iii) Molecular Weight: molecular weight of peptide (iv) Charge: net charge on peptide (v) Solubility: Good/Poor (vi) Motifs: Different motifs of length 3 and 4 present in the peptide (vii) Score_3 (viii) Score_4.

2.6 Summary

There are four major classes of antifungal agents that dominate the market, namely azoles, polyenes, echinocandins, and fluorinated pyrimidines. Due to an almost similar mechanism of action of available antifungal drugs, cross-antifungal resistance has taken place; as a result, antifungal drugs are becoming less effective. The increase in the number of patients with immuno-deficiency/immunosuppression-related diseases, resistance to existing antifungal compounds, and availability of limited therapeutic options have triggered the search for safer alternatives having enhanced features. In this direction, AFPs have emerged as promising novel antibiotic agents to treat fungal infections. The AFPs are produced by a diverse population of living organisms but identifying effective AFPs from natural sources is time-consuming and expensive. Wet lab researchers conduct various trials to identify novel AFPs from natural resources, which involves a lot of time and money. As a result, the *in-silico* tool is needed to undertake a preliminary screening of natural sources to discover potential AFPs. 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 (in the form of PESTV) with the 1DCNN-BiLSTM deep learning algorithm. To understand the contribution of transfer learning in the proposed framework, we performed the ablation studies by utilizing non-pretrained encodings from PAM250, BLOSUM62, and one-hot encoding (OHE) with a 1DCNN-BiLSTM deep learning algorithm. We also experimented with various machine-learning algorithms, and found that our proposed framework performed better. Our proposed model also obtained better generalization performance than the existing tools, which shows that wet-lab researchers can utilize our proposed model for identify-

ing novel AFPs from natural sources. By utilizing Deep-AFPpred, we identified AFPs from the ten antifungal proteins belonging to fungi of five different genera. The helical wheel representation of proposed AFPs showed that hydrophobic and hydrophilic residues are arranged in two different planes in most of the predicted AFPs. Therefore, they are more likely to adopt a helical structure in solution, which will be beneficial for the antifungal activity of these peptides. The identified AFPs can be synthesized as either linear or branched peptides, and their antifungal activity can be evaluated by similar techniques used by [53, 54] for bacterial pathogens. Similarly, the work may also be undertaken to develop hybrid peptides comprising two or more AFPs for achieving better activity, as has been demonstrated by [55] in the case of antibacterial peptides. We have also made a web server available at https://afppred.anvil.app/ to support researchers in identifying the novel AFPs from any protein. This web server is made flexible, wherein users can customize various parameters to identify AFPs from protein sequences. In the current work, we have adopted the criteria of probability, presence of unique motifs, and solubility in water for selecting promising AFPs. The AFPs selected utilizing these criteria may be chemically synthesized and tested for their activity against different fungal pathogens. In light of limited options for treating fungal infections and increasing antifungal drug resistance, the novel AFPs may be promising therapeutic molecules.

In the future, this work can be extended to build a two-stage classifier wherein the first stage will identify AFPs in protein sequences, and the second stage will provide information regarding the activity of identified peptides on different fungi. Although in the present work, we have considered all the AFPs available in the literature, there is still a scope to further enhance the performance of the proposed model by including more well-characterized AFPs that may be available in the future.