Computers and Electrical Engineering 123 (2025) 110098

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

# XAI-INVENT: An explainable artificial intelligence based framework for rapid discovery of novel antibiotics

Ritesh Sharma [a], Sameer Shrivastava [b,*], Sanjay Kumar Singh [c,*], Abhinav Kumar [c,d,**], Amit Kumar Singh [e], Sonal Saxena [f]

[a] Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, 147004, Punjab, India
[b] Facility for Research and Training on Bioassays and Biosensor, Veterinary Immunology Section, ICAR-Indian Veterinary Research Institute, Izatnagar, 243122, Uttar Pradesh, India
[c] Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221005, Uttar Pradesh, India
[d] School of Artificial Intelligence and Data Engineering, Indian Institute of Technology Ropar, Rupnagar, 140001, Punjab, India
[e] Department of Computer Science and Engineering, National Institute of Technology, Patna, 800004, Bihar, India
[f] Division of Veterinary Biotechnology, ICAR-Indian Veterinary Research Institute, Izatnagar, 243122, Uttar Pradesh, India

ARTICLE INFO

*Keywords:*
XAI
Antibiotics
Artificial intelligence
Critical
Bi-GRU
Bi-LSTM
Bi-TCN
1DCNN
Bacteriocin
Antibacterial
AMR
ESKAPEE
Antibacterial peptides

ABSTRACT

The failure of the most potent medicines to eradicate superbugs underscores the urgent need to develop new antimicrobial drugs. Antibacterial peptides (ABPs) are oligopeptides present in all multicellular organisms and serve as the first line of defense against pathogens. ABPs provide several benefits over conventional antibiotics; therefore, they have recently gained significant attention as an alternative. Finding ABPs in the laboratory is expensive and time-consuming. Therefore, wet-lab researchers use *in-silico* tools to discover ABPs from natural sources. The existing tools available for this purpose suffer from the limitation of being black boxes. In the present work, we developed XAI-INVENT, an explainable artificial intelligence-based framework for the rapid discovery of novel antibiotics. For building XAI-INVENT, first, the probability scores of deep learning models are fused, and then the fused scores are utilized with local interpretable model-agnostic explanations (LIME) for determining the critical amino acids. The value of performance metrics, namely Accuracy, Sensitivity, Precision, F1-Score, Specificity, and Matthews correlation coefficient obtained by the proposed framework for test data is ≈ 96 %, 96 %, 97 %, 96 %, 97 %, and 92 %, respectively. To help wet-lab researchers, XAI-INVENT is deployed as a web server at https://xai-invent.anvil.app/.

## 1. Introduction

Multiple mutations in the genome of microbes enable them to survive in the presence of drugs, resulting in antimicrobial resistance (AMR). Antibacterial peptides (ABPs) are molecules present in the immune system that protect the host from infections. ABPs provide many benefits over conventional antibacterial medications. They are naturally occurring, destroy bacteria in various ways, and have few adverse effects. Because of this, ABPs have recently received a lot of attention as an alternative to the available antibiotics [1–4]. As a consequence, peptide-based drugs account for 7% of all drugs authorized by the Food and Drug Administration

* Corresponding authors.
** Corresponding author at: Facility for Research and Training on Bioassays and Biosensor, Veterinary Immunology Section, ICAR-Indian Veterinary Research Institute, Izatnagar, 243122, Uttar Pradesh, India.
*E-mail addresses:* ritesh.sharma1@thapar.edu (R. Sharma), sameer.shrivastava@icar.gov.in (S. Shrivastava), sks.cse@iitbhu.ac.in (S.K. Singh), abhinav.rs.cse17@iitbhu.ac.in (A. Kumar), amit.singh@nitp.ac.in (A.K. Singh), sonal@icar.gov.in (S. Saxena).

https://doi.org/10.1016/j.compeleceng.2025.110098
Received 21 December 2023; Received in revised form 23 October 2024; Accepted 17 January 2025
Available online 12 February 2025

(FDA) in the previous five years [5]. Antimicrobial peptides have been shown to have therapeutic uses by the authors in [6]. The FDA has authorized seven AMP-based drugs so far, and a thorough analysis of those drugs can be found in [7].

The state of the art works [8–13] suffer from the limitation of being black boxes. Therefore, wet-lab researchers cannot interpret the reason behind the prediction and hence cannot use their domain expertise to modify the peptide sequence and confidently select the best ABP(s) for wet-lab synthesis. Therefore, adopting explainable AI (XAI) is crucial to uncover the reasons behind AI-driven decisions and promoting a deeper understanding of their implications. This not only addresses interpretability concerns but also improves the drug discovery process, facilitating more informed decision-making. The XAI has been used in different domains to enable users to understand the prediction made by the model [14–21]. Most of the works from distinct domains rely on the LIME algorithm for explainability. In [14] LIME algorithm is utilized for resource management for Enhanced Communication Efficiency in Hierarchical Federated Learning. In [15] LIME algorithm is utilized for hate speech detection and classification. In [16] LIME algorithm is utilized for white blood cell classification. In [20] LIME algorithm is utilized in the field of cybersecurity.

For better understanding, we can relate the task of classifying the peptide as ABP/Non-ABP to the sentiment analysis task, where, given a sentence, the goal is to determine whether the expressed opinion in the sentence is positive or negative. In the case of sentiment analysis, we can see that all words do not contribute equally in determining the sentiment of a sentence (some words have more contribution, some words have less contribution, and even some words have negative contribution). Consider the statement, ‘‘That was a horrible movie’’. The overall sentiment of this statement is negative. The word that contributed most towards this negative prediction is ‘‘horrible’’, while the remaining words contributed little or nothing. In the sentiment analysis task, identifying critical words is possible for humans, but identifying critical amino acids in the case of peptides is not at all possible.

Motivated by this, in the present work, we developed XAI-INVENT, an explainable artificial intelligence-based framework for rapid discovery of novel antibiotics. The practical implications and potential impact of XAI-INVENT on wet-lab research and antibiotic discovery is that $\approx$ 227 million protein sequences are available in the UniProt database [22], and even a single protein sequence can have multiple regions with antibacterial properties. Therefore, numerous ABPs can be identified from a single protein sequence. However, it is not feasible to synthesize and test all of them in vitro. Thus, in such cases, the knowledge of critical amino acids (amino acids responsible for the antibacterial nature of the sequence) enables wet-lab researchers to interpret the reason behind the prediction. This, in turn, will help them use their domain expertise to confidently select the best ABPs for wet-lab synthesis. Moreover, wet lab researchers can further narrow their selection based on their preferences of amino acids (like they may choose ABP in which their preferred amino acid is available as a critical amino acid).

The current work focuses on sequential data. The bidirectional gated recurrent unit (Bi-GRU), bidirectional long short-term memory (Bi-LSTM), bidirectional temporal convolutional network (Bi-TCN), and 1D convolutional Neural Network (1DCNN) are powerful neural network architectures that are designed to handle sequential data effectively. As a result, they are used in current work. Moreover, instead of relying on any one of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN, their decisions can be combined to make the ensemble classifier, which offers several benefits like (i) It aggregates the strengths of different models, which offers more confidence than the confidence offered by any single model. (ii) Reduces overfitting by balancing out individual model errors, making the ensemble more generalizable to new data. (iii) Improves robustness, making the model less sensitive to noise and outliers. Based on the benefits mentioned above, in the proposed framework XAI-INVENT, the probabilistic decisions of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN have been combined using soft voting to make the ensemble classifier. Then, the fused scores are utilized with local interpretable model-agnostic explanations (LIME) [23] for identifying critical amino acids.

To get more insights, we conducted a pilot study in which (i) We analyzed all the peptides and determined amino acids that are present as critical amino acids in most of the ABPs/Non-ABPs. (The top five critical amino acids for ABPs are W, K, R, D, and M, while E, D, T, S, and *N* are the top five critical amino acids for Non-ABPs). (ii) We analyzed some sample peptides that corresponded to each of the True Positive, True Negative, False Positive, and False Negative categories and identified critical amino acids (see Section 3.2.2). (iii) We have also shown that the ABP/Non-ABP can be transformed into Non-ABP/ABP by removing the critical amino acid(s) (see Section 3.2.3).

We also identified ABPs in bacteriocin obtained from the ESKAPEE group of bacteria. We chose the ESKAPEE group of bacteria because they are highly threatening, drug-resistant, World Health Organization (WHO) critical priority I and II pathogens that develop resistance mechanisms in response to environmental threat rapidly and thus pose a significant threat among the drug-resistant microbes [24,25].

The details of XAI-INVENT are shown in Fig. 1, which contains three major components: (i) **Extraction of peptides from proteins:** Substrings of length $\in$[5,50] are generated to obtain peptides from the input protein sequence. (ii) **Ensemble of base classifiers:** Corresponding to each peptide, the probability scores provided by Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN are fused using soft voting. (iii) **Explanation for the prediction using LIME:** Corresponding to each peptide, the fused score obtained from the ensemble classifier is provided to LIME, which provides a heatmap representing the critical amino acids in it. Wet lab researchers can combine the information from this heatmap with their domain expertise to make certain decisions.

The main contributions of our paper are summarized as follows:

1. We proposed an explainable artificial intelligence-based framework named XAI-INVENT for the rapid discovery of novel antibiotics.
2. To the best of our knowledge, proposed framework XAI-INVENT is the first of its kind, which not only identifies potent ABPs from protein sequences but also provides information about the amino acids that play an essential role in classifying a peptide.
3. Using XAI-INVENT, we identified ABPs in bacteriocin obtained from the ESKAPEE group of bacteria, which are highly threatening drug-resistant WHO priority I and II pathogens.

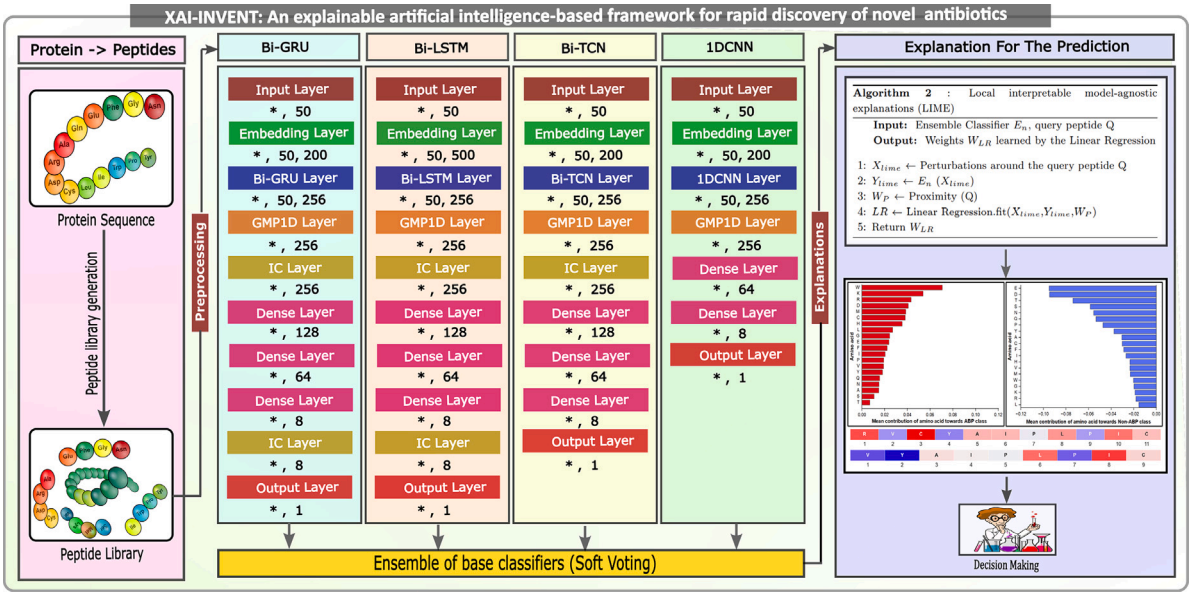



**Fig. 1.** Details of XAI-INVENT.

4. To help researchers find new ABPs from protein sequences, the model has been set up as a web server and is freely accessible online at https://xai-invent.anvil.app/.

The remaining sections of this work are organized as follows. Section 2 contains details about the dataset and the suggested framework. Section 3 presents the experiments and results. Identification of ABPs in bacteriocin obtained from the ESKAPEE is provided in Section 4. Section 5 provides the conclusion and future work.

## 2. Materials and methods

### 2.1. Dataset

We prepared dataset $D_s$ containing 20,648 peptides (ABPs: 10,324, Non-ABPs: 10,324). These peptides have length ∈ [5,50] and are made up of natural amino acids only. The ABPs, available in $D_s$, were obtained by compiling the data from the DBAASP v3 database. First, we queried the DBAASP v3 database for ABPs, which provided us with 131 097 ABPs. Then, the ABPs with non-natural amino acids were removed, and out of 131 097 ABPs, 100 089 ABPs were left. Afterward, the ABPs that had lengths of more than fifty and less than five were removed, and we were left with 93 625 sequences. After that, we removed the duplicate sequences and were left with 10 324 ABPs. Thereafter, an equal number of Non-ABPs were obtained from the Swiss-Prot database [26] using the similar approach used in the previous works [27–33]. Specifically, for obtaining Non-ABPs from Swiss-Prot, it was queried for the reviewed, manually annotated proteins that did not contain any of the following keywords: antioxidant, anticancer, defensin, bacteriocin, antiviral, anti-protist, antiendotoxin, antiparasitic, antitumor, antiprotozoal, antimicrobial, anti-inflammatory, antibiotic, antibacterial, anti-gram +, anti-gram-, insecticidal, cytokine, antimalarial, anti-MRSA, antifungal, anti-diabetic, anti-TB, anti-HIV, antibiofilm, anti-toxin, secreted, excreted, effector.

We further divided the $D_s$ into three sets, namely Training set ($S^{Train}$), Test set ($S^{Test}$), and Validation set ($S^{Val}$). $S^{Train}$ (defined by Eq. (1)) composed of around 60% peptides, $S^{Val}$ (defined by Eq. (2)) composed of around 20% peptides, and $S^{Test}$ (defined by Eq. (3)) composed of remaining 20% peptides.

$$
\begin{aligned}
&S^{Train} = S^{Train}_{ABPs} \cup S^{Train}_{Non-ABPs} \\
&\text{where,} \\
&S^{Train}_{ABPs} \cap S^{Train}_{Non-ABPs} = \emptyset \\
&|S^{Train}_{ABPs}| = 6195 \\
&|S^{Train}_{Non-ABPs}| = 6195 \\
&|S^{Train}| = 12390
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
&S^{Val} = S^{Val}_{ABPs} \cup S^{Val}_{Non-ABPs} \\
&\text{where,} \\
&S^{Val}_{ABPs} \cap S^{Val}_{Non-ABPs} = \emptyset \\
&|S^{Val}_{ABPs}| = 2065 \\
&|S^{Val}_{Non-ABPs}| = 2064 \\
&|S^{Val}| = 4129
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
&S^{Test} = S^{Test}_{ABPs} \cup S^{Test}_{Non-ABPs} \\
&\text{where,} \\
&S^{Test}_{ABPs} \cap S^{Test}_{Non-ABPs} = \emptyset \\
&|S^{Test}_{ABPs}| = 2064 \\
&|S^{Test}_{Non-ABPs}| = 2065 \\
&|S^{Test}| = 4129
\end{aligned} \tag{3}
$$

### 2.2. Proposed framework

In the current work, we proposed XAI-INVENT, an explainable artificial intelligence-based framework for the rapid discovery of novel antibiotics. For building XAI-INVENT, first, the probability scores provided by Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN deep learning algorithms are fused using a soft voting technique, and then the fused scores are utilized with LIME to identify the critical amino acids. We performed different experiments (related to the dimensionality of the embedding layer, learning rate, number of dense layers, number of filters, size of filter, number of neurons, etc.) while choosing the architecture of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN. This section provides the optimal architecture obtained for Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN after different experiments. The size of input and output corresponding to each layer of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN is shown in Fig. 1.

In Bi-GRU, the first layer is the Input layer, which we use to feed the data (peptides). Before feeding peptides, we performed the following pre-processing steps: (i) Mapped 20 amino acids to numerical values (K: 1, L: 2, R: 3, A: 4, G: 5, I: 6, V: 7, S: 8, T: 9, F: 10, P: 11, N: 12, E: 13, C: 14, Q: 15, D: 16, Y: 17, W: 18, H: 19, M: 20). (ii) Created all peptides of the same length (=50). Following this, the Embedding layer is used. This embedding layer learns the vector representation of each amino acid, which is present in the dictionary. The embedding layer can be initialized with pre-trained embeddings or random weights. In the current study, we utilized the pre-trained embeddings because they realize the concept of transfer learning, which helps this layer to learn better representation for each amino acid. These pre-trained embeddings were obtained using the word2vec technique (for each amino acid, word2vec learns a 200-length vector representation). Following the Embedding layer, we used a 256-neuron Bi-GRU layer (128 neurons in each GRU layer). In subsequent Section 2.3, we described the Bi-GRU layer, which forms the core parts of our proposed model. Following the Bi-GRU layer, we utilized the GlobalMaxPooling1D (GMP1D) layer, which accomplishes downsampling by taking the maximum value throughout the time dimension. Following the GMP1D layer, we employed the Independent component (IC) layer. The IC layer was introduced in [34], where the authors combined batch normalization with dropout. They conducted numerous trials and discovered that the IC layer leads to a more stable training process, quick convergence, and enhanced generalization. The output of the preceding IC layer is routed to dense layers $D_1$, $D_2$, and $D_3$, which contain 128, 64, and 8 neurons, respectively. Following that, we employed an IC layer, followed by an output layer comprised of one neuron.

In Bi-LSTM, the first layer is the Input layer, which we use to feed the peptides to the model. Following the input layer, the Embedding layer is used, which utilizes a 500-dimensional word2vec embedding vector. After the Embedding layer, we utilized a 256-neuron Bi-LSTM layer (128 neurons in each LSTM layer). In subsequent Section 2.4, we described the Bi-LSTM layer, which forms the core parts of our proposed model. After the Bi-LSTM layer, we used the GMP1D layer, followed by the IC layer. The output from the preceding IC layer is fed to dense layers $D_1$, $D_2$, and $D_3$ comprising 128, 64, and 8 neurons, respectively. Next, we used the IC layer followed by the output layer comprising one neuron.

In Bi-TCN, the first layer is the Input layer, which we use to feed the peptides to the model. Following the input layer, the Embedding layer is used, which utilizes a 200-dimensional word2vec embedding vector. After the Embedding layer, we utilized a 256-neuron Bi-TCN layer (128 neurons in each TCN layer). In subsequent Section 2.5, we described the Bi-TCN layer, which forms the core parts of our proposed model. After Bi-TCN layer, we used the GMP1D layer, followed by the IC layer. The output from the preceding IC layer is fed to dense layers $D_1$, $D_2$, and $D_3$ comprising 128, 64, and 8 neurons, respectively. Next, we used an output layer comprising one neuron.

In 1DCNN, the first layer is the Input layer, which we use to feed the peptides to the model. Following the input layer, the Embedding layer is used, which utilizes a 200-dimensional word2vec embedding vector. After the Embedding layer, we utilized a 256-neuron 1DCNN layer. In subsequent Section 2.6, we described the 1DCNN layer, which forms the core parts of our proposed model. After 1DCNN layer, we used the GMP1D layer. The output from the preceding GMP1D layer is fed to dense layers $D_1$ and $D_2$ comprising 64 and 8 neurons, respectively. Next, we used an output layer comprising one neuron.

We used the rectified linear unit (ReLU) activation function (defined by Eq. (4)) with each of Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN, and dense layer.

$$ReLU(x) = \begin{cases} 0, \text{ if } x \leq 0 \\ x, \text{ if } x > 0 \end{cases} \tag{4}$$

The sigmoid (Sig) activation function (defined by Eq. (5)) is used with the output layer of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN.

$$Sig(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

This output layer provides the probability score.

The optimizer used with each of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN is Adam, which uses exponential moving averages $r_t$ and $s_t$ of the gradient $g_t$ and $g_t^2$. Both $r_t$ and $s_t$ are initialized using zero vectors, and their values are updated with each iteration $t$. Adam's parameter update for each iteration $t$ is represented by Eqs. (6)–(11) .

$$g_t = \frac{\partial L}{\partial w}\Big|_{w_{t-1}} \tag{6}$$

$$r_t = \alpha r_{t-1} + (1 - \alpha) g_t \tag{7}$$

$$\hat{r}_t = \frac{r_t}{1 - \alpha^t} \tag{8}$$

$$s_t = \beta s_{t-1} + (1 - \beta) g_t^2 \tag{9}$$

$$\hat{s}_t = \frac{s_t}{1 - \beta^t} \tag{10}$$

$$w_t = w_{t-1} - \eta \left(\frac{\hat{r}_t}{\sqrt{\hat{s}_t} + \epsilon}\right) \tag{11}$$

where $\epsilon$ is used to handle division by zero condition, $\alpha$ and $\beta$ are used to handle the decay rates of $r_t$ and $s_t$, respectively, and $\eta$ ($= 10^{-5}$) denotes the learning rate. The EarlyStopping approach is used to deal with the problem of overfitting. This technique stops the training of the model when the validation loss starts increasing. We combined the predictions from the base classifiers utilized using the soft voting technique (Algorithm 1) to obtain the final predictions. The LIME algorithm (Section 2.8, Algorithm 2) is then utilized with this fused score for determining the critical amino acids.

**Algorithm 1** : Ensemble of base classifiers

**Input:** Base classifiers (Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN), query peptide Q.
**Output:** Class C (= 0 for Non-ABP and 1 for ABP) and
Probability P (∈ [0,1]) corresponding to ABP
for query peptide Q

/* Probability values ( $P_1, .., P_4$ ) are provided by Models ( Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN ) for query sequence $Q$ */

$P_1 \leftarrow$ Bi-GRU (Q)
$P_2 \leftarrow$ Bi-LSTM (Q)
$P_3 \leftarrow$ Bi-TCN (Q)
$P_4 \leftarrow$ 1DCNN (Q)
P $\leftarrow \frac{(P_1+P_2+P_3+p_4)}{4}$
**if** $P > 0.5$ **then**
    C←1
**else**
    C←0
**end if**
Return $C$, $P$

**Algorithm 2** : Local interpretable model-agnostic explanations (LIME)

**Input:** Ensemble Classifier $E_n$, query peptide Q
**Output:** Weights $W_{LR}$ learned by the Linear Regression

$X_{lime} \leftarrow$ Perturbations around the query peptide Q
$Y_{lime} \leftarrow E_n\ (X_{lime})$
$W_P \leftarrow$ Proximity (Q)
$LR \leftarrow$ Linear Regression.fit($X_{lime}$,$Y_{lime}$,$W_P$)
Return $W_{LR}$

### 2.3. Gated recurring units

The GRU is formed up of repeating cells. Cell accepts $v_t$ and $h_{t-1}$ as inputs and produces $h_t$ at each time step $t$ ($1 \leq t \leq 50$). Each cell's computing steps are described by the Eqs. (12)–(15)

$$u_t = sigmoid(W_u x_t + b_u) \tag{12}$$

$$s_t = sigmoid(W_s x_t + b_s) \tag{13}$$

$$c_t = tanh(W_c[s_t \otimes h_{t-1}, v_t] + b_c) \tag{14}$$

$$h_t = (1 - u_t) \otimes h_{t-1} + u_t \otimes c_t \tag{15}$$

In the above equations, $\otimes$ denotes the element-wise multiplication operation, $\|$ denotes the concatenate operator, $v_t$ represent embedding vector, $x_t$ provides the result obtained by concatenating $h_{t-1}$ and $v_t$, $u_t$ represent update gate, $s_t$ represent reset gate, $c_t$ represent candidate state, ($W_u$, $b_u$) represent the weights and biases of the update gate, ($W_s$, $b_s$) represent the weights and biases of the reset gate, ($W_c$, $b_c$) represent the weights and biases of the candidate state.

### 2.4. Long short-term memory

The LSTM is formed up of repeating cells. At each time step $t$ ($1 \leq t \leq 50$), the input to the cell is $v_t$, $h_{t-1}$ and $c_{t-1}$ and it provides $h_t$, $c_t$ as output. Each cell's computing steps are described by the Eqs. (16)–(20):

$$i_t = sigmoid(W_i x_t + b_i) \tag{16}$$

$$f_t = sigmoid(W_f x_t + b_f) \tag{17}$$

$$c_t = (f_t \otimes c_{t-1}) \oplus (i_t \otimes tanh(W_c x_t + b_c)) \tag{18}$$

$$o_t = sigmoid(W_o x_t + b_o) \tag{19}$$

$$h_t = o_t \otimes tanh(c_t) \tag{20}$$

In the above equations, $\otimes$ denotes the element-wise multiplication operation, $\oplus$ denotes the element-wise addition operation, $v_t$ represent embedding vector, $x_t$ provides the result obtained by concatenating $h_{t-1}$ and $v_t$, $i_t$ represent input gate, $f_t$ represent forget gate, $c_t$ represent cell state, $o_t$ represent output gate, $h_t$ represent hidden state, ($W_i$, $b_i$) represent the weights and biases of $i_t$, ($W_f$, $b_f$) represent the weights and biases of $f_t$, ($W_c$, $b_c$) represent the weights and biases of $c_t$, ($W_o$, $b_o$) represent the weights and biases of $o_t$.

### 2.5. Temporal convolutional network

TCNs [35–40] are deep learning architectures that are created by altering Convolutional Neural Networks (CNNs). TCN, as opposed to ordinary convolution, employs dilated casual convolution. The following are some TCN-related concepts:

(i) **Causal convolutional**:

For getting the output at time t, the input from time 0 to t is considered (ii) **Dilations**: The receptive field scales linearly with depth, which is the major disadvantage of causal convolutional. This requires the employment of multiple layers to extend the receptive field. This can be fixed by combining dilations (skipping values between convolutional inputs) with casual convolutions. (iii) **Receptive Field**: The maximum number of steps a filter can take back in time. Let $D(i)$ signify the dilation corresponding to layer $i$, $F(i)$ represent the number of time steps that the dilated causal convolution layer $i$ can perceive, $s$ denote the kernel size, and $n$ represents the number of dilations. The receptive field $F(n)$ (defined by Eq. (26)) for dilated causal convolutions can be calculated as follows:

$$F(n) = F(n-1) + (s-1)D(n) \tag{21}$$

$$F(n-1) = F(n-2) + (s-1)D(n-1) \tag{22}$$

$$F(2) = F(1) + (s-1)D(2) \tag{23}$$

$$F(1) = 1 + (s-1)D(1) \tag{24}$$

Substituting Eqs. (22)–(24) in Eq. (21):

$$F(n) = 1 + (s-1)(D(1) + \cdots + D(n-1) + D(n)) \tag{25}$$

If dilation increases exponentially then D(1) = $2^0$, D(2) = $2^1$, D(3) = $2^2$, …., D(n-1) = $2^{n-2}$, D(n) = $2^{n-1}$. Substituting these values in Eq. (25) we get:

$$\begin{aligned} F(n) &= 1 + (s-1)(2^0 + 2^1 + 2^2 + \cdots + 2^{n-2} + 2^{n-1}) \\ &= 1 + (s-1)(2^n - 1) \end{aligned} \tag{26}$$

Residual blocks are utilized in TCN to avoid vanishing and exploding gradient problems. Each block has two identical dilated causal convolutions. The block's outputs are obtained by adding the final convolution results to the inputs.

### 2.6. 1D convolutional neural networks

The convolution operation is the basis of 1D convolutional neural networks (1DCNN). CNN was designed initially to detect spatial patterns in images. This was performed by moving a small window horizontally and vertically across an image, followed by a convolution operation. In the case of peptides, the convolution operation necessitates that the window evaluates the entire embedding vector of each amino acid. As a result, the window's width becomes fixed, and it can only move in one direction. As a result, the convolution operation is referred to as a 1D convolution operation, and CNN is referred to as a 1DCNN.

### 2.7. Ensemble of base classifiers

Wet-lab researchers employ in-silico tools to select peptides for wet-lab synthesis and analysis. Thus, even marginal enhancements in the tool's efficacy can save their time and money. The BiGRU, BiLSTM, BiTCN, and 1DCNN algorithms are heterogeneous. Therefore, peptides misclassified by the classifiers corresponding to any of these algorithms may be accurately classified by others. Moreover, the resultant prediction obtained by combining the predictions from these diverse classifiers can enhance overall performance. Motivated by this, we ensembled the predictions of BiGRU, BiLSTM, BiTCN, and 1DCNN algorithms using a soft voting algorithm (Algorithm 1). This soft voting algorithm considers the probability scores of each base model and calculates the weighted average of these probabilities to make the final prediction.

### 2.8. Local interpretable model-agnostic explanations (LIME)

LIME stands for local interpretable model-agnostic explanations. This technique faithfully explains any black-box model's predictions by approximating it locally with an interpretable model. To do so, LIME first generates a new dataset by executing random perturbations around the query point. After this, predictions are made for the newly created dataset using the black-box model (for which interpretability is required). In addition to prediction, the newly created samples are weighted according to their closeness to the query point. Then, an interpretable linear model is trained, and the weights learned are used to unbox the black-box model for the query point. The algorithm for LIME is given by Algorithm 2. Corresponding to each amino acid of the peptide, LIME associates a weight value w ∈[−ve, 0, +ve]. If w is +ve, then it means that amino acid contributes towards +ve prediction (ABP). If w is −ve, then it means that amino acid is contributing towards −ve prediction (Non-ABP). For better visualization, these weights are shown as a heatmap where +ve values are represented by red and −ve values are represented by blue. Our hypothesis is that if the LIME is working perfectly, then the weightage of red-colored amino acids will be more than the blue-colored amino acids for the sequences that are predicted as ABP (The red-colored amino acids are known as critical amino acids in this case because these amino acids are responsible for the prediction of sequence as ABP), whereas the weightage of blue colored amino acids must be greater than red colored amino acids for the sequences which are predicted as Non-ABP (The blue colored amino acids are known as critical amino acids in this case because these amino acids are responsible for the prediction of sequence as Non-ABP). To verify the correctness of LIME, we performed different experimentations as shown in Sections 3.2.2 and 3.2.3.

## 3. Experiments and results

The results obtained from various experiments conducted before finalizing the ensemble classifier $E_n$ are presented in this section. This section also includes the outcomes of different experiments we carried out using LIME. To implement XAI-INVENT, we utilized the Keras library with Tensorflow as the backend. We used various performance metrics defined by Eqs. (27)–(32) to access the performance. These metrics are described in terms of the following:

- True Positive (TP): Peptides correctly classified as ABPs.
- True Negative (TN): Peptides correctly classified as Non-ABPs.
- False Positive (FP): Peptides misclassified as ABPs.
- False Negative (FN): Peptides misclassified as Non-ABPs.

**Accuracy** ($A_{cc}$): Ratio of accurately classified peptides to the total number of peptides.

$$A_{cc} = \frac{TN + TP}{FP + TN + TP + FN} \tag{27}$$

**Table 1**
Result obtained for $S^{Val}$ from Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN.

| Algorithm | $A_{cc}$ (%) | $S_n$ (%) | $P_r$ (%) | $F_s$ (%) | $S_p$ (%) | MCC (*100) |
|---|---|---|---|---|---|---|
| Bi-GRU | 95.37 | 94.91 | 95.79 | 95.35 | 95.83 | 90.75 |
| Bi-LSTM | 95.22 | 94.62 | 95.78 | 95.20 | 95.83 | 90.46 |
| Bi-TCN | 95.64 | 95.69 | 95.59 | 95.64 | 95.59 | 91.28 |
| 1DCNN | 95.03 | 95.10 | 94.97 | 95.03 | 94.96 | 90.07 |

**Table 2**
Results obtained for $S^{Val}$ from the ensemble classifiers constructed by combining any two of the Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms.

| Algorithm | $A_{cc}$ (%) | $S_n$ (%) | $P_r$ (%) | $F_s$ (%) | $S_p$ (%) | MCC (*100) |
|---|---|---|---|---|---|---|
| LG-SV | 95.44 | 94.76 | 96.07 | 95.41 | 96.12 | 90.90 |
| LT-SV | 96.00 | 95.54 | 96.43 | 95.98 | 96.46 | 92.01 |
| LC-SV | 95.64 | 94.72 | 96.49 | 95.60 | 96.56 | 91.29 |
| GT-SV | 96.02 | 95.73 | 96.29 | 96.01 | 96.31 | 92.05 |
| GC-SV | 95.81 | 95.35 | 96.23 | 95.79 | 96.26 | 91.62 |
| TC-SV | 95.49 | 95.35 | 95.62 | 95.48 | 95.63 | 90.99 |

**Table 3**
Result obtained for $S^{Val}$ from the ensemble classifiers constructed by combining any three of the Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms.

| Algorithm | $A_{cc}$ (%) | $S_n$ (%) | $P_r$ (%) | $F_s$ (%) | $S_p$ (%) | MCC (*100) |
|---|---|---|---|---|---|---|
| LGT-SV | 96.19 | 95.49 | 96.85 | 96.17 | 96.89 | 92.40 |
| LGC-SV | 96.24 | 95.35 | 97.09 | 96.21 | 97.14 | 92.50 |
| LTC-SV | 95.90 | 95.30 | 96.47 | 95.88 | 96.51 | 91.82 |
| GTC-SV | 95.93 | 95.35 | 96.47 | 95.90 | 96.51 | 91.86 |

**Sensitivity** ($S_n$): Correct prediction ratio of positive samples. It is also known as recall or TPR.

$$S_n = \frac{TP}{FN + TP} \tag{28}$$

**Precision** ($P_r$): Ratio of accurately classified ABPs to the total peptides that were classified as ABPs. It is also known as the positive predictive value (PPV).

$$P_r = \frac{TP}{TP + FP} \tag{29}$$

**F1-Score** ($F_s$): Harmonic mean of $S_n$ and $P_r$.

$$F_s = \frac{2 \times S_n \times P_r}{S_n + P_r} \tag{30}$$

**Specificity** ($S_p$): Ratio of accurately classified Non-ABPs to the actual Non-ABPs present in the dataset. It is also known as true negative rate (TNR).

$$S_p = \frac{TN}{TN + FP} \tag{31}$$

**Matthews correlation coefficient (MCC)**: Provides the correlation between actual and predicted values and is regarded as the best metric to evaluate the performance of the model on an imbalanced dataset.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TN + FP) \times (TP + FN) \times (TP + FP) \times (TN + FN)}} \tag{32}$$

### *3.1. Results from various experiments performed before arriving at ensemble classifier $E_n$*

We developed an ensemble classifier $E_n$ by employing a soft voting technique with the Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN deep learning algorithms. Before arriving at this decision, we conducted the following experiments: (i) Build classifiers using Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms taking one at a time (ii) Build ensemble classifiers by combining 2 of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms (iii) Build ensemble classifiers by combining 3 of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms (iv) Build ensemble classifiers by combining all Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms. For conducting the aforementioned experiments $S^{Train}$ was used as training data, and $S^{Val}$ was used as validation data.

The results obtained from Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN taking one at a time are shown in Table 1 and their comparison in terms of composite metrics is provided in Fig. 2. As can be seen from this figure, Bi-TCN performed the best.

The results obtained by combining any 2 of Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN algorithms using soft voting (Where LG-SV, LT-SV, LC-SV, GT-SV, GC-SV, TC-SV represent soft voting used with (Bi-LSTM and Bi-GRU), (Bi-LSTM and Bi-TCN), (Bi-LSTM and 1DCNN), (Bi-GRU and Bi-TCN), (Bi-GRU and 1DCNN), (Bi-TCN and 1DCNN), respectively) are shown in Table 2 and their comparison in terms of composite metrics is provided in Fig. 3. As can be seen from this figure, GT-SV performed the best.

**Table 4**
Result obtained for $S^{Val}$ from the ensemble classifier constructed by combining Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms.

| Algorithm | $A_{cc}$ (%) | $S_n$ (%) | $P_r$ (%) | $F_s$ (%) | $S_p$ (%) | MCC (*100) |
|---|---|---|---|---|---|---|
| LGTC-SV | 96.29 | 95.39 | 97.14 | 96.26 | 97.18 | 92.60 |

**Table 5**
Results obtained for $S^{Test}$ by the selected ensemble classifier (LGTC-SV).

| Algorithm | $A_{cc}$ (%) | $S_n$ (%) | $P_r$ (%) | $F_s$ (%) | $S_p$ (%) | MCC (*100) |
|---|---|---|---|---|---|---|
| Bi-GRU | 95.08 | 94.52 | 95.59 | 95.05 | 95.64 | 90.17 |
| Bi-LSTM | 95.34 | 95.00 | 95.65 | 95.33 | 95.69 | 90.70 |
| Bi-TCN | 95.54 | 95.63 | 95.45 | 95.54 | 95.44 | 91.08 |
| 1DCNN | 95.34 | 95.00 | 95.65 | 95.33 | 95.69 | 90.70 |
| LGTC-SV | 96.17 | 95.63 | 96.66 | 96.15 | 96.70 | 92.35 |

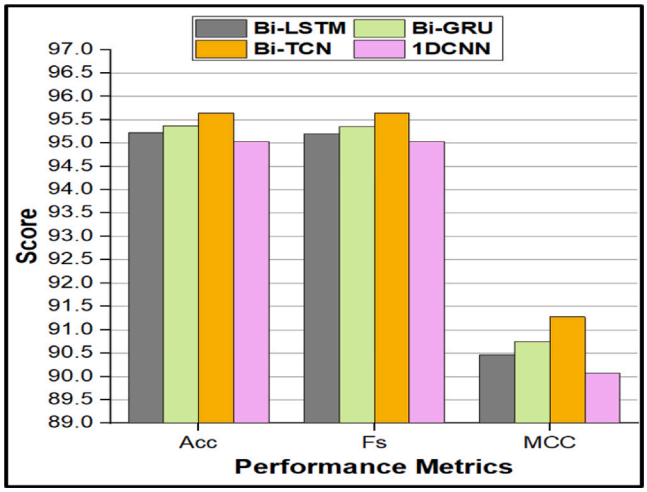



**Fig. 2.** Comparison of results obtained from Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN taking one at a time.

The results obtained by combining any 3 of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms using soft voting (Where LGT-SV, LGC-SV, LTC-SV, GTC-SV represent soft voting used with (Bi-LSTM, Bi-GRU, and Bi-TCN), (Bi-LSTM, Bi-GRU, and 1DCNN), (Bi-LSTM, Bi-TCN, and 1DCNN), (Bi-GRU, Bi-TCN, and 1DCNN), respectively) are shown in Table 3 and their comparison in terms of composite metrics are provided in Fig. 4. As can be seen from this figure, LGC-SV performed the best.

The results obtained by combining all Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms using soft voting are shown in Table 4 (LGTC-SV)

The best results from all the cases (Bi-TCN, GT-SV, LGC-SV and LGTC-SV) are compared in Fig. 5. As can be seen from this figure, LGTC-SV performed the best. Therefore, we have considered this as our classifier in the proposed framework.

$S^{Test}$ was not utilized for hyperparameter tuning or training. Therefore, it was used to access the actual performance of LGTC-SV, and the results obtained are provided in Table 5. As can be seen from this table, the performance obtained by LGTC-SV for $S^{Test}$ is almost similar to that obtained for $S^{Val}$. The reason for the better performance of $E_n$ and similar performance attained by $E_n$ for both $S^{Test}$ and $S^{Val}$ is its stability (The deep learning algorithms which are utilized in $E_n$ are heterogeneous. As a result, the peptide that got misclassified by one model got correctly classified by another. Therefore, the aggregate result of multiple models is less noisy than that of the individual model, leading to model stability and robustness).

### *3.2. Results from various experiments conducted using LIME*

#### *3.2.1. Global predictions based on LIME*

We applied LIME to peptides in $D_s$ and identified amino acids that are present as critical amino acids in most of (i) ABPs: obtained by calculating the mean of positive weights attained by each amino acid (see Fig. 6). (ii) Non-ABPs: obtained by calculating the mean of negative weights attained by each amino acid (see Fig. 7). The top five critical amino acids for ABPs are W, K, R, D, and M, while E, D, T, S, and N are the top five critical amino acids for Non-ABPs. We discovered that certain amino acids are working as critical amino acids in both ABPs and Non-ABPs (like D is present in the top 5 critical amino acids for both ABPs and Non-ABPs). The reason for this can be understood by considering the following two sentences from the sentiment analysis task: (i) This movie

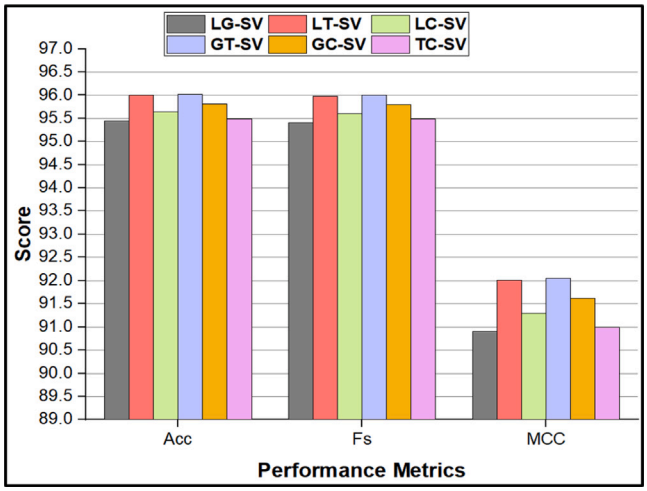



**Fig. 3.** Comparison of results obtained by combining any two of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN.

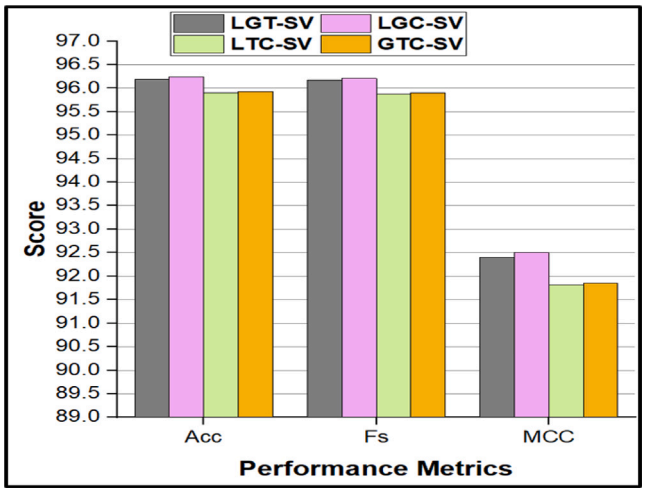



**Fig. 4.** Comparison of results obtained by combining any three of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN.

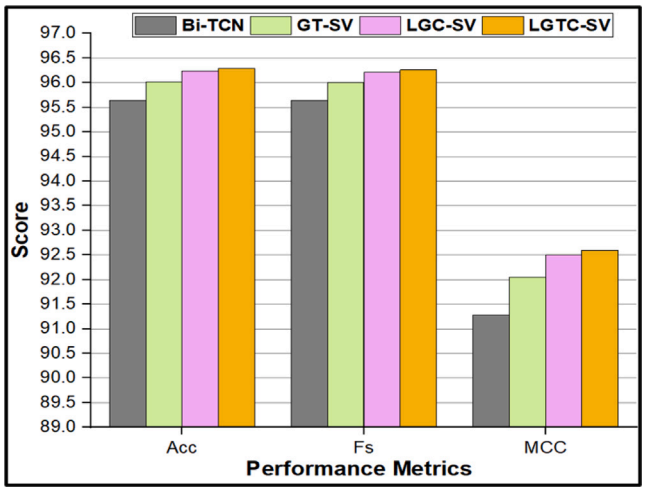



**Fig. 5.** Comparison of the best results obtained for all the cases.

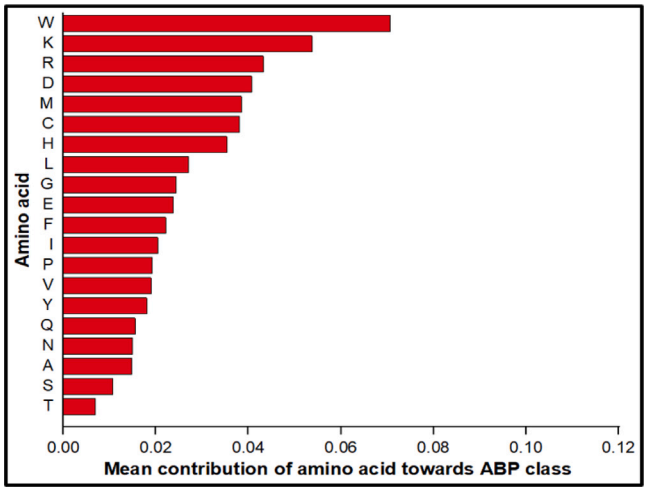



**Fig. 6.** Mean contribution of amino acid towards ABP class.

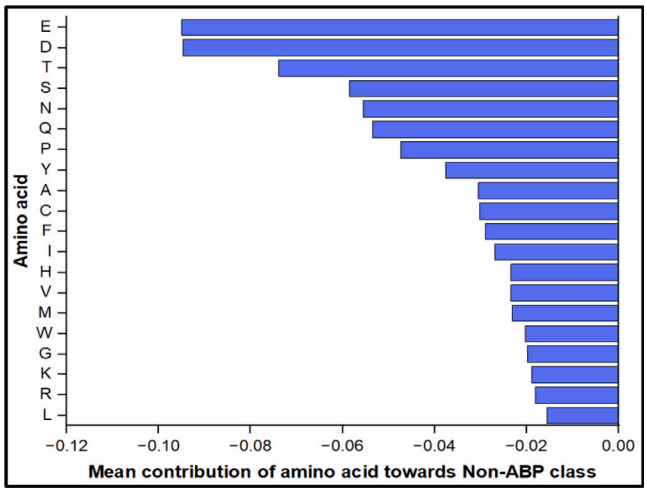



**Fig. 7.** Mean contribution of amino acid towards Non-ABP class.

is not bad, and (ii) This movie is not good. The word "not" contributes towards positive sentiment in (i), whereas it contributes towards negative sentiment in (ii).

### *3.2.2. LIME for representative peptides*

The predictions made by LIME for sample peptides from each of the True Positive, True Negative, False Positive and False Negative categories are shown in Fig. 8(A), (B), (C) and (D), respectively. From Fig. 8(A), we can see that the weightage of red-colored amino acids is high compared to blue-colored amino acids. Therefore, the model has classified the peptide as ABP. Similarly, Fig. 8(C) shows the high weightage of red-colored amino acids compared to blue-colored amino acids, leading to the wrong prediction of Non-ABP as ABP.

From Fig. 8(B), we can see that the weightage of blue-colored amino acids is high compared to red-colored amino acids. Therefore, the model has classified the peptide as Non-ABP. Similarly, Fig. 8(D) shows the high weightage of blue-colored amino acids compared to red-colored amino acids, leading to the wrong prediction of ABP as Non-ABP.

### *3.2.3. Transformation of ABP/Non-ABP to Non-ABP/ABP*

By eliminating red-colored amino acid(s) (responsible for moving the peptide prediction towards ABP), the ABP can be converted to Non-ABP. Fig. 8(A) shows a heatmap of predictions made by LIME for sample ABP, and Fig. 9(A) shows a heatmap of predictions made by LIME for the same ABP after removing critical amino acid (s). The critical amino acids removed from 1st peptide are R and C, present at positions 1 and 3, respectively. The critical amino acids removed from 2nd peptide are H, L, and K, present at positions 8, 9, and 11, respectively. The critical amino acid removed from the 3rd peptide is W, present at position 1. The critical

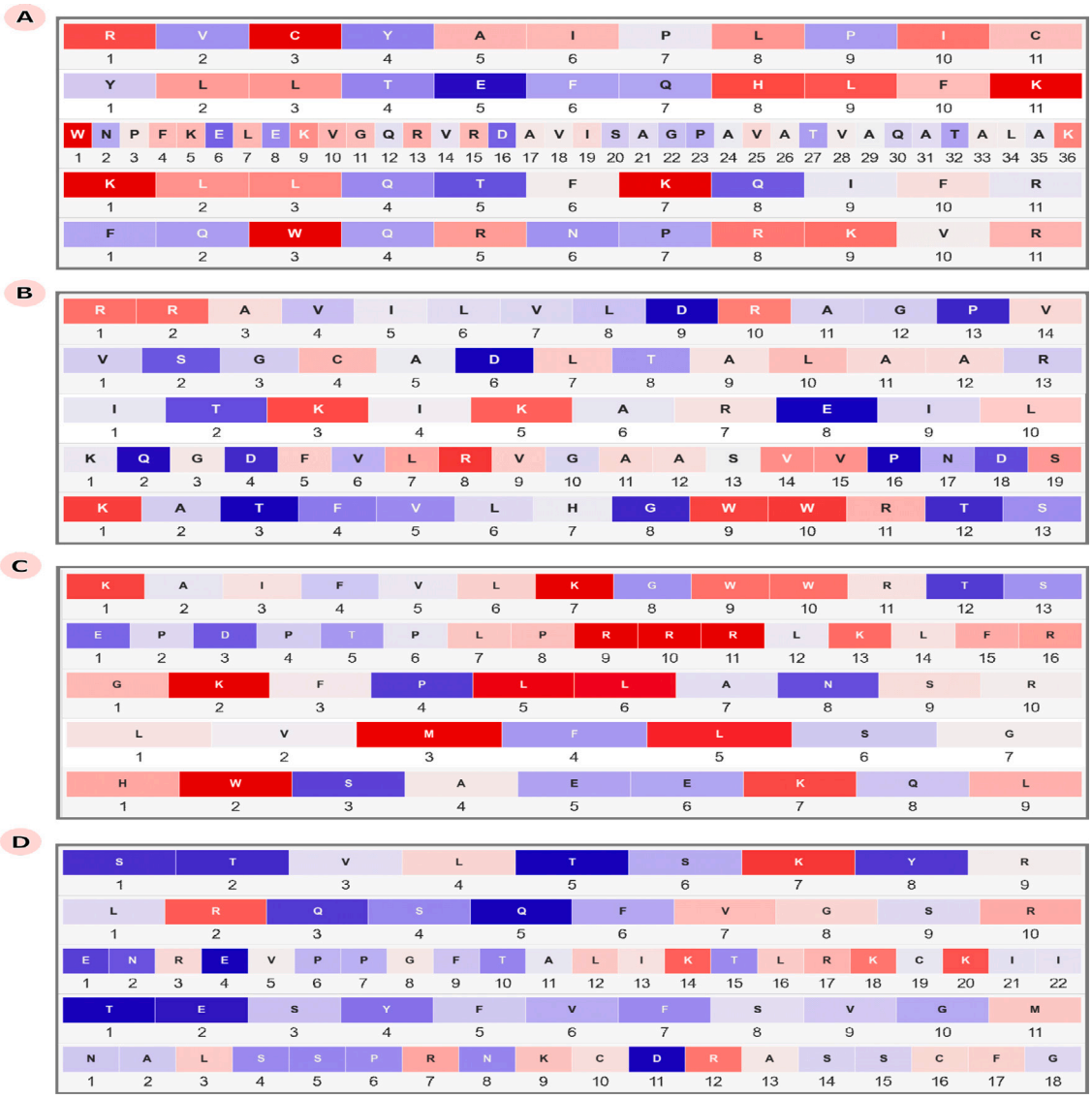



**Fig. 8.** Predictions made by LIME for sample peptides from each of (A) True Positive, (B) True Negative, (C) False Positive and (D) False Negative categories.

amino acids removed from the 4th peptide are K, and K, present at positions 1, and 7, respectively. The critical amino acid removed from the 5th peptide is W, which is present at position 3. The probability values obtained from the ensemble classifier $E_n$ for these five peptides before and after removing the aforementioned critical amino acid(s) are (0.98, 0.85, 0.93, 0.94, 0.98) and (0.39, 0.06, 0.40, 0.13, 0.27), respectively. These probability scores (for each of the five peptides, initially, the probability score was greater than o.5, and after the removal of critical amino acids, the probability score for each of the peptides became more than 0.5) show the transformation of ABPs to Non-ABPs on the removal of critical amino acid(s).

Similarly, by eliminating the blue-colored amino acid(s) (responsible for moving the peptide prediction towards Non-ABP), the Non-ABP can be converted to ABP. Fig. 8(B) shows a heatmap of predictions made by LIME for sample Non-ABP, and Fig. 9(B) shows a heatmap of predictions made by LIME for the same Non-ABP after removing critical amino acid (s). The critical amino acids removed from 1st peptide are D and P, present at positions 9 and 13, respectively. The critical amino acids removed from the 2nd peptide are S and D, present at positions 2 and 6, respectively. The critical amino acids removed from the 3rd peptide are T and E, present at positions 2 and 8, respectively. The critical amino acids removed from the 4th peptide are Q, D, P, and D, present at positions 2, 4, 16, and 18, respectively. The critical amino acids removed from the 5th peptides are T, G, and T, present at positions 3, 8, and 12, respectively. The probability values obtained from the ensemble classifier $E_n$ for these five peptides before and after removing the aforementioned critical amino acid(s) are (0.31, 0.16, 0.36, 0.25, 0.08) and (0.97, 0.93, 0.99, 0.95, 0.94), respectively. These probability scores (for each of the five peptides, initially, the probability score was less than o.5, and after the

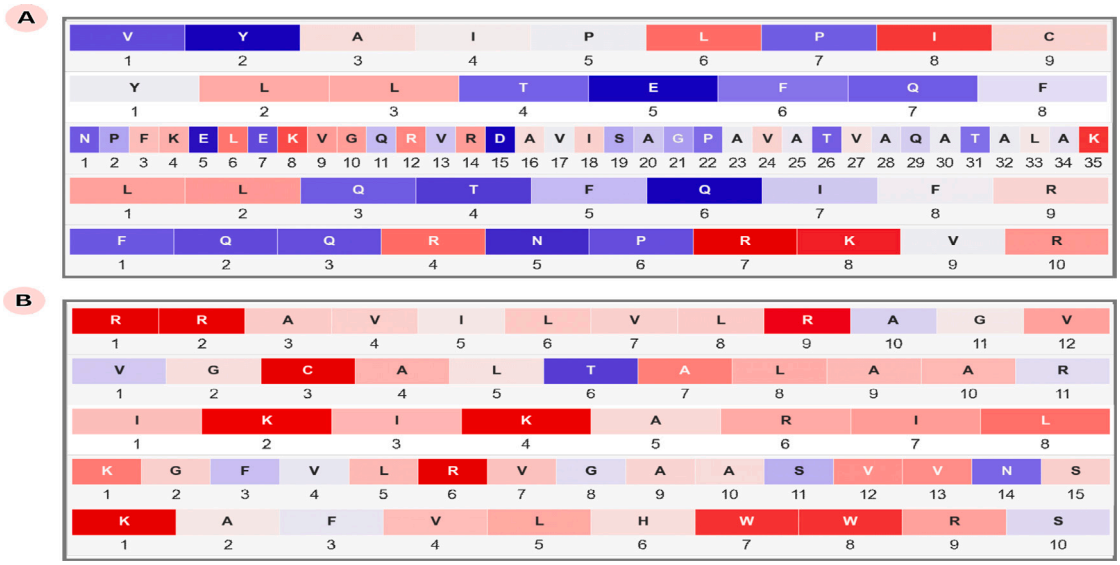



**Fig. 9.** (A) Conversion of ABP to Non-ABP after removing critical amino acid (s) (B) Conversion of Non-ABP to ABP after removing critical amino acid (s).

removal of critical amino acids, the probability score for each of the peptides became more than 0.5) show the transformation of Non-ABPs to ABPs on the removal of the critical amino acid(s).

## 4. Prediction of ABPS in the bacteriocin obtained from the ESKAPEE group of bacteria

Corresponding to bacteriocin obtained for each bacteria from the ESKAPEE group (details are provided in Table 6), we proposed one ABP using the following steps: (i) Utilized XAI-INVENT, which provided probability score and heatmap corresponding to each ABP identified from the protein sequences. (ii) Considered the ABPs from the previous step which have length $\in$ [20, 30] and then sorted them in decreasing order of probability (iii) Obtained the helical wheel representation (generated using the HeliQuest web server [41] available at https://heliquest.ipmc.cnrs.fr/) and structure (obtained from PEP-FOLD3 web server [42] available at https://bioserv.rpbs.univ-paris-diderot.fr/services/PEP-FOLD3/) for the ABPs obtained from the previous step (iv) Identified the topmost ABP from the previous step, which displays good helical wheel representation and structure both. (v) Proved the novelty of the ABP obtained from the previous step. For this, we first performed the protein–protein BLAST (Blastp) of the peptide identified in (iv) with respect to the available ABPs. Then, we used the commonly used alignment tool accessible at https://www.uniprot.org/align/ to align the highest-ranked ABP produced by Blastp with the corresponding query ABP. We noticed that 100% alignment is not there, confirming the uniqueness of our peptide.

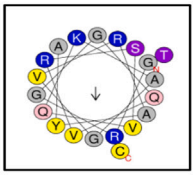
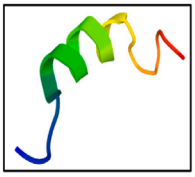
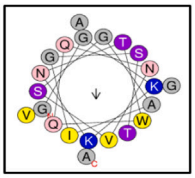
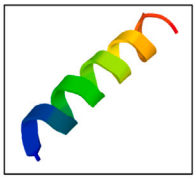
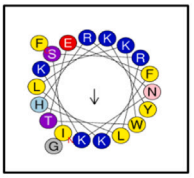
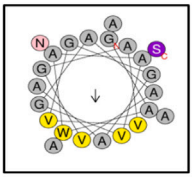
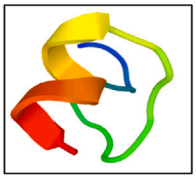
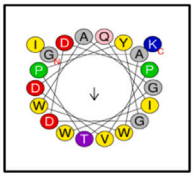
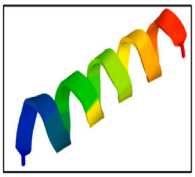
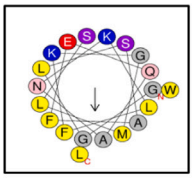
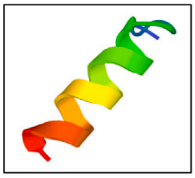
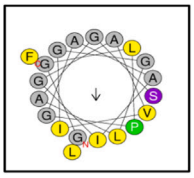
To aid wet-lab researchers, we made XAI-INVENT available as a web app at https://xai-invent.anvil.app/. The web server accepts as input the protein sequence and length $\in$ [5, 50] (on the basis of which the peptide library is generated) and returns ABPs with probability scores and a heatmap produced from LIME for the top peptide. Fig. 10 shows the output from the web server for the sample bacteriocin protein of Enterococcus faecium.

## 5. Conclusion and future perspective

The failure of antibiotics underscores the necessity for the development of new antimicrobial medications. Antimicrobial peptides (ABPs) are present in all multicellular animals and offer various advantages over conventional antibacterial drugs. Therefore, they have garnered considerable attention as an alternative . However, the process of identifying ABPs from natural sources is time-consuming and labor-intensive. Thus, wet-lab researchers use *in-silico* tools to detect ABPs. But, the current tools available for this purpose are black boxes. Therefore, in this study, we introduce XAI-INVENT, an explainable artificial intelligence-based framework for the rapid discovery of novel antibiotics. To construct XAI-INVENT, we first combine the probability scores generated by deep learning algorithms using a soft voting technique. Subsequently, these combined scores are fed into the LIME algorithm to identify the critical amino acids. The performance metrics of the proposed framework on test data, namely $A_{cc}$, $S_n$, $P_r$, $F_s$, $S_p$, and MCC, are approximately 96, 96, 97, 96, 97, and 92, respectively. Using the proposed model, we detected ABPs in the bacteriocin of the ESKAPEE group of bacteria. Additionally, to assist wet-lab researchers, we have made XAI-INVENT accessible as a web application at https://xai-invent.anvil.app/.

Antibiotic resistance is a severe worldwide health concern that reduces the effectiveness of conventional medications against common bacterial infections. ABPs offer numerous advantages over traditional antibiotics, which has led to a significant surge in

**Table 6**
Antibacterial peptides derived from bacteriocin of the ESKAPEE group of bacteria and proposed for wet-lab synthesis and experimentation.

| Antibacterial peptides | Helical Wheel | Predicted Structure |
|---|---|---|
| Bacteriocin: Enterococcus faecium [OTO49785.1] — Proposed ABP: GRGGQVRSVQKAGVRAYATC<br>Source of Existing ABP: Synthetic — Existing ABP: ACYCRIPACIAGERRYATCIYQGRLWAFCC |  |  |
| Bacteriocin: Staphylococcus aureus [OIV48016.1] — Proposed ABP: GGKKNTWQQNVSGAIGSTVAGA<br>Source of Existing ABP: Hymenochirus boettgeri — Existing ABP: IKVPAIVKDTQKKVAKGVISTVADALSKS |  |  |
| Bacteriocin: Klebsiella pneumoniae [WP_109223240.1] — Proposed ABP: ISRLHRNKKKWTEFKLKYGF<br>Source of Existing ABP: Synthetic — Existing ABP: GLLKKWLKKWKEFKRIVGY |  |  |
| Bacteriocin: Acinetobacter baumannii [TPV09527.1] — Proposed ABP: GAWAAVGAAVGAVVGGAAAAANS<br>Source of Existing ABP: Synthetic — Existing ABP: KKAAAVAAAAAVAAWAAVAAAKKKK |  |  |
| Bacteriocin: Pseudomonas aeruginosa [AVZ37966.1] — Proposed ABP: GAWWAGTPYGDDPVDQIWIK<br>Source of Existing ABP: Synthetic — Existing ABP: ILPWKWAWWAWRR |  |  |
| Bacteriocin: Enterobacter quasiroggenkampii [WP_235404758.1] — Proposed ABP: GGLSAFEQANKLFKGMLSWL<br>Source of Existing ABP: Synthetic — Existing ABP: KKLFKKILKWL |  |  |
| Bacteriocin: Escherichia coli [UJY38014.1] — Proposed ABP: GGLPGAAIGAVIGGLAGSLF<br>Source of Existing ABP: Bombina maxima — Existing ABP: ILGPVIKTIGGVIGGLLKNL |  |  |

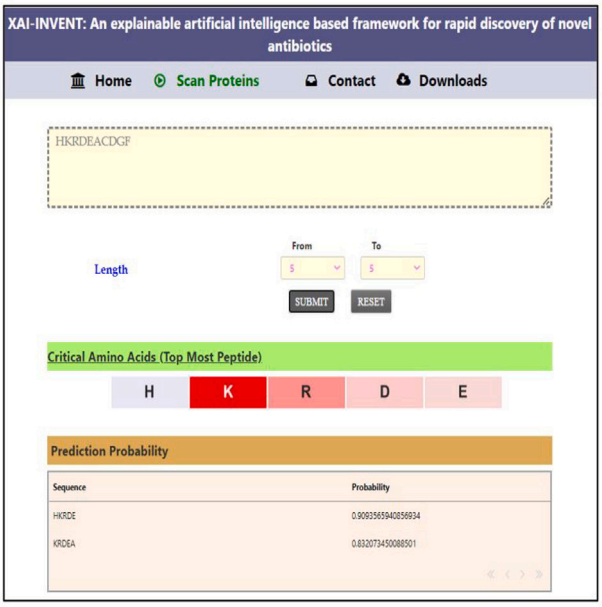



**Fig. 10.** Web server corresponding to XAI-INVENT.

attention towards them as viable alternatives to existing antibiotic options. As a result, the amount of experimentally validated ABP-related data is steadily expanding. The main drawback of the proposed system is its inability to learn continually from new data. As a result, it will become obsolete with time. Therefore, the proposed framework can be improved in the future by incorporating the concept of continual learning, in which a model continuously learns from ABP-related data over time. This will enable the framework to incrementally acquire, update, and exploit knowledge throughout its lifetime and prevent it from becoming obsolete over time.

Another limitation of the current study is that bacteria-specific information is not considered. As a result, the proposed framework will not provide the effectiveness of the identified ABP on the particular bacteria (i.e., at what concentration will the ABP be effective on specific bacteria). In the future, this work can be expanded to create a two-stage cascade model. The first stage will identify the novel ABP, while the second stage will provide information about the concentration at which the identified ABP will be effective on specific bacteria.

The current work can be enhanced in the future by incorporating a more thorough analysis of the following : (i) **Positional importance:** Examining whether the ABP classification exhibits positional preferences (e.g., if beginning or ending peptide segments are more influential). (ii) **Higher-level structure:** Although the current analysis concentrates on the contributions at the amino acid level, the examination of sub-sequence structures could provide more valuable insights. (iii) **Automated transformation:** The development of a deep learning-based approach to identify the minimal number of mutations necessary to convert non-ABPs to ABPs (and vice versa).

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The support and resources provided by National Agricultural Science Fund under the project entitled “Developing novel therapeutic strategies for mitigating antimicrobial resistance” sanctioned vide F.No. NASF/BGAM-9006-2022-23 dated 25-05-2022 at ICAR-IVRI and IIT(BHU) are thankfully acknowledged.

## Data availability

Data will be made available on request.

## References


[1] Lei J, Sun L, Huang S, Zhu C, Li P, He J, Mackey V, Coy DH, He Q. The antimicrobial peptides and their potential clinical applications. Am J Transl Res 2019;11(7):3919.
[2] Huan Y, Kong Q, Mou H, Yi H. Antimicrobial peptides: classification, design, application and research progress in multiple fields. Front Microbiol 2020;11:2559.
[3] Zhang Q-Y, Yan Z-B, Meng Y-M, Hong X-Y, Shao G, Ma J-J, Cheng X-R, Liu J, Kang J, Fu C-Y. Antimicrobial peptides: mechanism of action, activity and clinical potential. Mil. Med. Res. 2021;8:1–25.
[4] Xuan J, Feng W, Wang J, Wang R, Zhang B, Bo L, Chen Z-S, Yang H, Sun L. Antimicrobial peptides for combating drug-resistant bacterial infections. Drug Resist Updat 2023;100954.
[5] Torre BGDL, Albericio F. Peptide therapeutics 2.0. Molecules 2020;25(10). URL https://europepmc.org/articles/PMC7287585.
[6] Datta S, Roy A. Antimicrobial peptides as potential therapeutic agents: A review. 2020, http://dx.doi.org/10.1007/s10989-020-10110-x.
[7] Chen CH, Lu TK. Development and challenges of antimicrobial peptides for therapeutic applications. Antibiotics 2020;9(1).
[8] Shao J, Zhao Y, Wei W, Vaisman II. AGRAMP: machine learning models for predicting antimicrobial peptides against phytopathogenic bacteria. Front Microbiol 2024;15:1304044.
[9] Yan J, Zhang B, Zhou M, Campbell-Valois F-X, Siu SW. A deep learning method for predicting the minimum inhibitory concentration of antimicrobial peptides against Escherichia coli using Multi-Branch-CNN and Attention. Msystems 2023;8(4):e00345–23.
[10] Bajiya N, Choudhury S, Dhall A, Raghava GP. AntiBP3: A method for predicting antibacterial peptides against gram-positive/negative/variable bacteria. Antibiotics 2024;13(2):168.
[11] Lin T-T, Yang L-Y, Lu I-H, Cheng W-C, Hsu Z-R, Chen S-H, Lin C-Y. AI4AMP: an antimicrobial peptide predictor using physicochemical property-based encoding method and deep learning. Msystems 2021;6(6):e00299–21.
[12] Burdukiewicz M, Sidorczuk K, Rafacz D, Pietluch F, Chilimoniuk J, Rödiger S, Gagat P. Proteomic screening for prediction and design of antimicrobial peptides with AmpGram. Int J Mol Sci 2020;21(12):4310.
[13] Szymczak P, Możejko M, Grzegorzek T, Jurczak R, Bauer M, Neubauer D, Sikora K, Michalski M, Sroka J, Setny P, et al. Discovering highly potent antimicrobial peptides with deep generative model HydrAMP. Nat Commun 2023;14(1):1453.
[14] Patni S, Lee J. Explainable AI empowered resource management for enhanced communication efficiency in hierarchical federated learning. Comput Electr Eng 2024;117:109260.
[15] Kibriya H, Siddiqa A, Khan WZ, Khan MK. Towards safer online communities: Deep learning and explainable AI for hate speech detection and classification. Comput Electr Eng 2024;116:109153.
[16] Bhatia K, Dhalla S, Mittal A, Gupta S, Gupta A, Jindal A. Integrating explainability into deep learning-based models for white blood cells classification. Comput Electr Eng 2023;110:108913.
[17] Rocha A, Monteiro M, Mattos C, Dias M, Soares J, Magalhães R, Macedo J. Edge AI for internet of medical things: A literature review. Comput Electr Eng 2024;116:109202.
[18] Ghasemieh A, Kashef R. Towards explainable artificial intelligence in deep vision-based odometry. Comput Electr Eng 2024;115:109127.
[19] Alani MM. An explainable efficient flow-based industrial IoT intrusion detection system. Comput Electr Eng 2023;108:108732.
[20] Sharma DK, Mishra J, Singh A, Govil R, Srivastava G, Lin JC-W. Explainable artificial intelligence for cybersecurity. Comput Electr Eng 2022;103:108356.
[21] Kumar A, Tiwari H, Singh R, Singh AK, Singh SK. SLIDE-Net: A sequential modeling approach with adaptive fuzzy C-mean empowered data balancing policy for IDC detection. IEEE Trans Fuzzy Syst 2024;32(10):5557–70.
[22] UniProt: the universal protein knowledgebase in 2023. Nucleic Acids Res 2023;51(D1):D523–31.
[23] Ribeiro MT, Singh S, Guestrin C. " why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016, p. 1135–44.
[24] Yu Z, Tang J, Khare T, Kumar V. The alarming antimicrobial resistance in ESKAPEE pathogens: Can essential oils come to the rescue? Fitoterapia 2020;140:104433.
[25] Reza A, Sutton JM, Rahman KM. Effectiveness of efflux pump inhibitors as biofilm disruptors and resistance breakers in gram-negative (ESKAPEE) bacteria. Antibiotics 2019;8(4):229.
[26] Consortium U. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res 2019;47(D1):D506–15.
[27] Lata S, Mishra NK, Raghava GP. AntiBP2: improved version of antibacterial peptide prediction. BMC Bioinformatics 2010;11(1):1–7.
[28] Singh V, Shrivastava S, et al. StaBle-ABPpred: a stacked ensemble predictor based on biLSTM and attention mechanism for accelerated discovery of antibacterial peptides. Brief Bioinform 2021. http://dx.doi.org/10.1093/bib/bbab439, bbab439.
[29] Sharma R, Shrivastava S, Kumar Singh S, Kumar A, Saxena S, Kumar Singh R. Deep-ABPpred: identifying antibacterial peptides in protein sequences using bidirectional LSTM with word2vec. Brief Bioinform 2021. http://dx.doi.org/10.1093/bib/bbab065.
[30] Singh V, Shrivastava S, Kumar Singh S, Kumar A, Saxena S. Accelerating the discovery of antifungal peptides using deep temporal convolutional networks. Brief Bioinform 2022;23(2).
[31] Sharma R, Shrivastava S, et al. AniAMPpred: artificial intelligence guided discovery of novel antimicrobial peptides in animal kingdom. Brief Bioinform 2021. http://dx.doi.org/10.1093/bib/bbab242.
[32] Sharma R, Shrivastava S, et al. Deep-AFPpred: identifying novel antifungal peptides using pretrained embeddings from seq2vec with 1DCNN-BiLSTM. Brief Bioinform 2021. http://dx.doi.org/10.1093/bib/bbab422.
[33] Sharma R, Shrivastava S, Singh SK, Kumar A, Singh AK, Saxena S. Deep-AVPpred: Artificial intelligence driven discovery of peptide drugs for viral infections. IEEE J Biomed Health Inf 2021;1. http://dx.doi.org/10.1109/JBHI.2021.3130825.
[34] Chen G, Chen P, Shi Y, Hsieh C-Y, Liao B, Zhang S. Rethinking the usage of batch normalization and dropout in the training of deep neural networks. 2019, arXiv preprint arXiv:1905.05928.
[35] Jothi ESJ, Anitha J, Hemanth DJ. A photoplethysmography-based diagnostic support system for obstructive sleep apnea using deep learning approaches. Comput Electr Eng 2022;102:108279.
[36] Niu K, Zhou Y, Lu G, Tai W, Zhang K. PMCT: Parallel multiscale convolutional temporal model for MOOC dropout prediction. Comput Electr Eng 2023;112:108989.
[37] Sun H, Xia M, Hu Y, Lu S, Liu Y, Wang Q. A new sorting feature-based temporal convolutional network for remaining useful life prediction of rotating machinery. Comput Electr Eng 2021;95:107413.
[38] Wei L, Li B, Li Y, Zhu Y. Time interval aware self-attention approach for knowledge tracing. Comput Electr Eng 2022;102:108179.
[39] Sharma R, Shrivastava S, Singh SK, Kumar A, Singh AK, Saxena S. Artificial intelligence-based model for predicting the minimum inhibitory concentration of antibacterial peptides against ESKAPEE pathogens. IEEE J Biomed Health Inf 2023;1–10. http://dx.doi.org/10.1109/JBHI.2023.3271611.
[40] Sharma R, Shrivastava S, Singh SK, Kumar A, Singh AK, Saxena S. EnDL-HemoLyt: Ensemble deep learning-based tool for identifying therapeutic peptides with low hemolytic activity. IEEE J Biomed Health Inf 2023;1–11. http://dx.doi.org/10.1109/JBHI.2023.3264941.
[41] Gautier R, Douguet D, Antonny B, Drin G. HELIQUEST: a web server to screen sequences with specific $\alpha$-helical properties. Bioinformatics 2008;24(18):2101–2.
[42] Lamiable A, Thévenet P, Rey J, Vavrusa M, Derreumaux P, Tufféry P. PEP-FOLD3: faster de novo structure prediction for linear peptides in solution and in complex. Nucleic Acids Res 2016;44(W1):W449–54. http://dx.doi.org/10.1093/nar/gkw329.