Briefings in Bioinformatics, 2025, **26(1)**, bbaf066
https://doi.org/10.1093/bib/bbaf066
Problem Solving Protocol

ESM-BBB-Pred: a fine-tuned ESM 2.0 and deep neural networks for the identification of blood–brain barrier peptides

Ansar Naseem[1], Fahad Alturise[2], Tamim Alkhalifah[3]*, Yaser Daanial Khan[4]

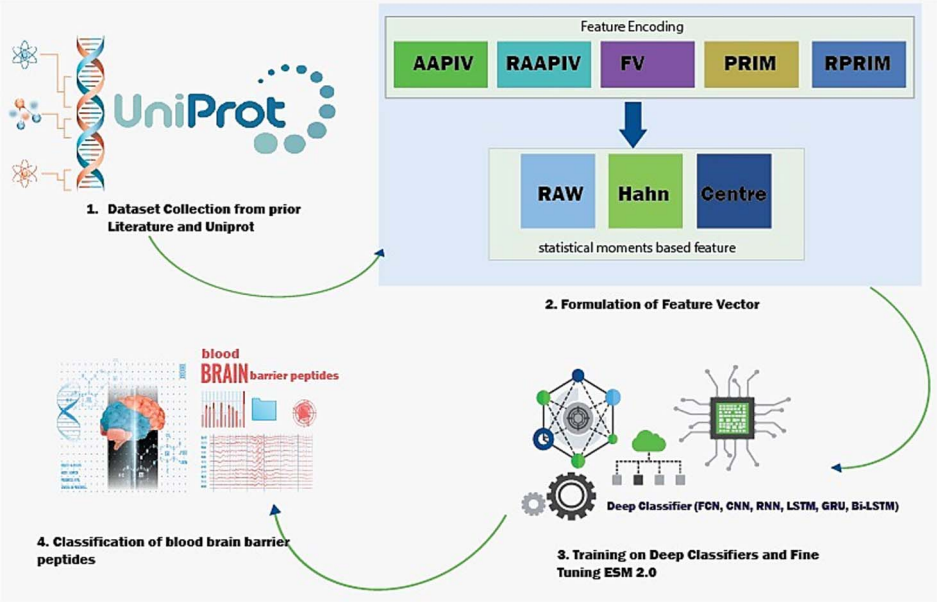
[1]Department of Software Engineering, Superior University, 17 KM Raiwind Road Lahore, Punjab 55150, Pakistan
[2]Department of Cybersecurity, College of Computer, Qassim University, Buraydah, Saudi Arabia
[3]Department of Computer Engineering, College of Computer, Qassim University, Buraydah, Saudi Arabia
[4]Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore, Pakistan

*Corresponding author. Department of Computer Engineering, College of Computer, Qassim University, Buraydah, Saudi Arabia. E-mail: tkhliefh@qu.edu.sa

Abstract

Blood–brain barrier peptides (BBBP) could significantly improve the delivery of drugs to the brain, paving the way for new treatments for central nervous system (CNS) disorders. The primary challenge in treating CNS disorders lies in the difficulty pharmaceutical agent's face in crossing the BBB. Almost 98% of small molecule drugs and nearly all large molecule drugs fail to penetrate the BBB effectively. Thus, identifying these peptides is vital for advancements in healthcare. This study introduces an enhanced intelligent computational model called BBB-PEP- Evolutionary Scale Modeling (ESM), designed to identify BBBP. The relative positions, reverse position and statistical moment-based features have been utilized on the existing benchmark dataset. For classification purpose, six deep classifiers such as fully connected networks, convolutional neural network, simple recurrent neural networks, long short-term memory (LSTM), bidirectional LSTM, and gated recurrent unit have been utilized. In addition to harnessing the effectiveness of the pre-trained model, a protein language model ESM 2.0 has been fine-tuned on a benchmark dataset for BBBP classification. Three tests such as self-consistency, independent set testing, and five-fold cross-validation have been utilized for evaluation purposes with evaluation metrics includes accuracy, specificity, sensitivity, and Matthews correlation coefficient. The fine-tuned model ESM 2.0 has shown superior results as compared to employed classifiers and surpasses the existing benchmark studies. This system will support future research and the scientific community in the computational identification of BBBP.

Graphical Abstract



Keywords: artificial intelligence 1; machine learning 2; computational biology 3; Proteomics 4; Computational Biology 5; Bioinformatics 6; Deep Learning 7; blood-brain barrier 8; peptides 9

Received: June 25, 2024. **Revised:** January 24, 2025. **Accepted:** February 7, 2025

Introduction

All of the body's tissues and organs depend on blood arteries for the transportation of oxygen and nutrients [1]. The blood arteries that vascularized the central nervous system (CNS) have specific properties identified as the blood–brain barrier [2]. This barrier controls ion, chemical, and cell flow between the circulation and the brain. By carefully regulating CNS homeostasis, the blood–brain barrier protects neural tissue from damaging poisons and viruses and promotes normal neuronal function. Changes in this barrier's integrity perform a major function in the development and course of a number of neurological illnesses [3]. Controlling these processes depends on the existence of barrier layers at crucial points where the blood and brain tissue converge [1].

Blood–brain barrier penetrating peptides (BBBPs) can cross the blood–brain barrier via a variety of methods without damaging the structure's integrity [4]. Certain BBBPs have been shown in studies to promote medication transport into the brain, presenting new possibilities for the creation of treatments for diseases of the CNS [5]. The stagnation in treating CNS disorders arises from the significant challenge of pharmacological drugs crossing the BBB. Almost 100% of large molecule-based medicines and 98% of small molecule-based drugs are unable to effectively penetrate the BBB [6].

The proposed study has made following contributions

1) PRIM, RPRIM, AAPIV, RAAPIV, and FV were computed to construct a feature vector on a collected benchmark dataset.
2) For dimensionality reduction, Hahn, Raw, and central based Statistical moments have been utilized
3) Deep architectures such as fully connected networks (FCN), convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory networks (LSTM), bidirectional LSTM (Bi-LSTM), and gated recurrent unit (GRU) have been employed for prediction of BBB peptides.
4) A pre-trained model Evolutionary Scale Modeling (ESM) 2.0 has been fine-tuned on a benchmark dataset.
5) Three tests such as self-consistency, independent set, and five-fold cross-validation (CV) have been utilized for prediction of BBB peptides.
6) Accuracy, specificity, recall, MCC and AUC score have been employed as evaluation metrics for BBB peptides prediction.

There has only been a few research on the computational discovery of blood–brain barrier peptides. A study by Dai *et al.*[7] accomplished a study for identification of BBB peptides that utilizes feature selection approach to eliminate repetitive and irrelevant variables. Finally, Logistic Regression (LR) was used to predict BBB peptides [7]. The next work has extended the dataset and utilized several feature descriptors for feature vector computation [8]. For classification purposes, this study has employed classifiers such as Decision Tree (DT), Random Forest (RF), LR, Support Vector Classifier (SVC), Gaussian Naive Bayes, XGBoost (XGB) and K-nearest neighbor (KNN) [8]. Another study by Chen *et al.* [9] has extended the benchmark dataset and utilizes the feature descriptors such as CKSAAP and PAAC. The classifiers such as DT, KNN, RF, AdaBoost, LogisBoost, GentleBoost, rbfSVM, and linearSVM have been employed for BBB peptides identification [9]. We have contributed to the latest work on BBB peptides prediction by utilizing the ensemble approaches such as bagging classifiers consisting of RF and Extra Trees (ET), boosting consists of LGBM and XGB. Finally, stacking employed XGB and ET utilizes and blending ensemble employed utilizes the LGBM and RF as base learners and LR as Meta learner [10].

Materials and methods

This section states about the methodology which comprises dataset description, feature vector approaches and classification approaches. The classifier section includes six Deep architectures: FCN, CNN, RNN, Bi-LSTM, LSTM and GRU. Finally, a pre-trained protein language model ESM2.0 has been employed.

Figure 1 depicts the architecture used to identify BBBPs. The relative positioning and statistical moments-based features have been employed. Six deep learning approaches have been used.

Dataset description

The benchmark dataset was compiled by Chen *et al.* [9]. Several research works, including B3Pdb Kumar *et al.* [11], Dorpe *et al.* [12], and publicly available datasets from BBPpred Dai *et al.* [7] and B3Pred Kumar *et al.* [8], were the data sources of the BBBPs that underwent experimental validation. The peptides related to the blood–brain barrier, brain, Brainpeps, B3Pdb, venom, toxins, trans membrane transport, transfer, membranes, neuro, and hemolysis, were removed from the sequences gathered from UniProt using exact query parameters for the collection of non-BBBPs. Dai *et al.* [7] then used CD-HIT to eliminate redundant sequences with a 10% sequence identity cut-off. Peptide sequences containing ambiguous residues were eliminated. This procedure results in generating 425 non-BBBPs. There are 425 positive and 425 negative samples in the entire dataset. The feature vectors were created from a pooled dataset of positive and negative sequences. To achieve better results, three classifiers' hyper parameters were tuned over the full dataset. After determining the best hyper parameters, the dataset was split into 77% for training and 23% for testing.

Feature formulation

Peptides sequences have processed using a feature extraction approach which utilizes composition-specific and position specific features. These popular methods consist of the following components.

Position relative incidence matrix

The intrinsic properties of proteins are largely dependent on where amino acid residues are positioned throughout the polypeptide chain. A matrix is assembled to capture positional correlations through all residues to highlight intricate patterns established by the arrangement of residues [13]. In order to evaluate the position-specific information of a protein, the 20 unique amino acid residues that are found throughout every polypeptide chain are taken into consideration while creating the PRIM matrix [14].

$$R_{PRIM} = \begin{bmatrix} R_{1\to1} & R_{1\to2} & \cdots & R_{1\to y} & \cdots & R_{1\to20} \\ R_{2\to1} & R_{2\to2} & \cdots & R_{2\to y} & \cdots & R_{2\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{x\to1} & R_{x\to2} & \cdots & R_{x\to y} & \cdots & R_{i\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{A\to1} & R_{A\to2} & \cdots & R_{A\to y} & \cdots & R_{A\to20} \end{bmatrix} \quad (1)$$

Every element of the PRIM matrix (Rij) represents the sum which is derived from the ith residue's relative placement in relation to the jth residue, showing that the ith residue is present at that place. The resultant matrix has 400 coefficients as a result. Statistical moments are computed in order to reduce the

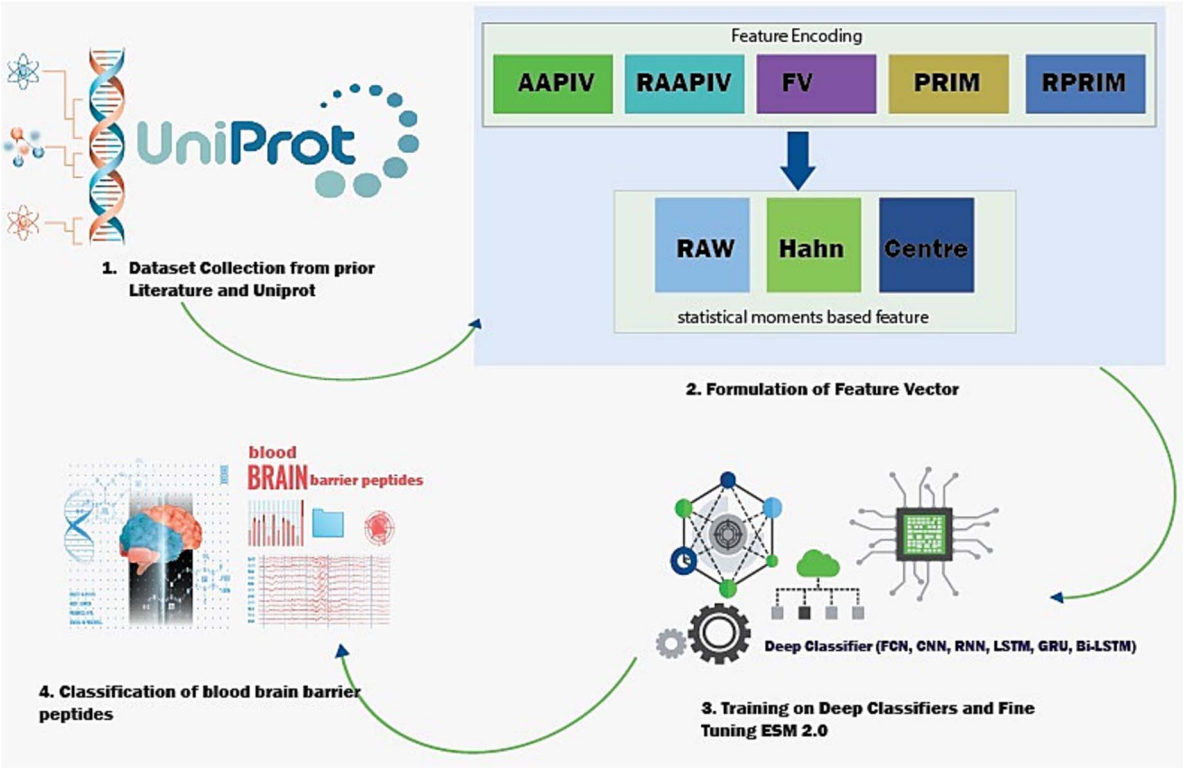


Figure 1. Architecture for BBBPs identification.

dimension complexity; this generates a list of thirty features that are taken out of the 400-coefficient matrix that was initially created.

Reverse position relative incidence matrix

Similar to the previous enumeration process, the RPRIM method investigates to uncovering hidden characteristics of homologous peptide sequences. RPRIM is calculated using the original sequence's reverse sequence [15]. The RPRIM matrix that was produced by this process is displayed below.

$$R_{RPRIM} = \begin{bmatrix} Q_{1\to1} & Q_{1\to2} & \cdots & Q_{1\to y} & \cdots & Q_{1\to20} \\ Q_{2\to1} & Q_{2\to2} & \cdots & Q_{2\to y} & \cdots & Q_{2\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ Q_{x\to1} & Q_{x\to2} & \cdots & Q_{x\to y} & \cdots & Q_{i\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ Q_{A\to1} & Q_{A\to2} & \cdots & Q_{A\to y} & \cdots & Q_{A\to20} \end{bmatrix} \quad (2)$$

The RPRIM matrix possesses the same dimensionality as the PRIM matrix, with 400 coefficients. Nevertheless, as with PRIM, the feature dimension of RPRIM is decreased to 30 coefficients by using statistical moments.

Frequency vector

For the purpose of figuring out residue distribution across a polypeptide chain in a specific sequence, the frequency vector has been used as a helpful tool [16]. It calculates the frequency of a given protein's residue. Protein sequence composition and distribution data are assured due to the FV feature. The FV is depicted in the following.

$$FV = [f_1, f_2, f_3, \ldots, f_{20}] \quad (3)$$

Based on the alphabetic ordinal value of each residue in the peptide sequence, the FV, a 20-dimensional vector, calculates its frequency.

Accumulative absolute position incidence vector

The FV gathers the distribution-specific information of every residue in amino acid and recognizes perplexing features in a protein's composition. Conversely, the relative positions of the residues are not recognized by the FV. The development of the AAPIV divided relative position-specific information into four quarters in order to address this problem [17]. The presence of the 20 native amino acids, as listed below, is the basis for this data.

$$K = [\forall_1, \forall_2, \forall_3, \ldots, \forall_n] \quad (4)$$

Where the AAPIV's i_{th} segment is determined as

$$\forall_i = \Sigma^n{}_{k=1}\ \beta_k \quad (5)$$

Regarding a particular residue, k indicates a random place. The AAPIV's specified part, represented as I, aggregates all the locations where the i_{th}nucleotide is present.

Reverse accumulative absolute position incidence vector

The primary difference between RAAPIV and AAPIV is that it constructs the output vector by utilizing the reverse sequence of the original sample. The reverse facilitates the retrieval of

supplementary information on position-specific information, hence allowing the discovery of profound and concealed characteristics within the sequences [18]. The representation of the vector is as follows.

$$RAAPIV = [n_1, n_2, n_3, , \ldots, n_m] \quad (6)$$

Statistical moments

The raw, Hahn, and central moments of the genomic data are used to populate the feature set, which contribute key features to the model's input vector. Researchers have demonstrated that the relative positions and contents of bases in proteomic and genomic sequences influence their characteristics. In order to improve the feature vector, computational and mathematical techniques have been established to capture the correlated arrangement of nucleotide bases in genomic sequences [19]. Having a strong emphasis on associated placement is essential to creating a robust and comprehensive feature set [20].

As Hahn moments necessitate two-dimensional data, genomic sequences are converted into a two-dimensional matrix S′ with dimensions k∗k. While this matrix S′ retains the same information as the original matrix S′, consequently.

$$\mathbf{k} = \sqrt{\mathbf{n}} \quad (7)$$

$$S' = |S_{11}\ S_{12} \ldots S_{1n}\ S_{21\ldots}\ S_{22\ldots} \ldots S_{2n\ldots}\ S_{n1}\ S_{n2} \ldots S_{nn}\ | \quad (8)$$

The generated square matrix is used to compute statistical moments, which produce a fixed-size feature vector and reduce dimensionality.

The raw moments of order a + b can be calculated using the equation below.

$$U_{ab} = \Sigma^{n}_{e=1} \Sigma^{n}_{f=1}\ e^a f^b \delta ef \quad (9)$$

Important information is embedded in the sequences' moments, notably up to the third order areU_{00}, U_{10},U_{11},U_{20}, U_{02}, U_{21}, U_{12},U_{03} and U_{30}. To calculate the central moments $(\bar{x}, \bar{y})$, to find the central point of the data, first compute the centroid. After that, the central moments are calculated using this centroid and the steps listed below:

$$\upsilon_{ab} = \Sigma^{n}_{e=1} \Sigma^{n}_{f=1} \left(e - \bar{x}\right)^a \left(f - \bar{y}\right)^b \delta ef \quad (10)$$

A square grid is the discrete input used to calculate Hahn moments. This method makes it possible to rebuild the original data using inverse Hahn moments, which highlights the regularity and reversibility of the data. The reversibility characteristic of Hahn moments ensures that the information from the original sequences is preserved and incorporated into the model through the feature vector. The following formula illustrates how Hahn moments are calculated.

$$h_n^{x,y}(p, Q) = (Q + V - 1)_n (Q - 1)_n \times \Sigma^{n}_{z=0} (-1)^z \times \frac{(-n)_z (-p)_z (2Q + x + y - n - 1)_z}{(Q + y - 1)_z (Q - 1)_z} \frac{1}{z!} \quad (11)$$

The equation makes use of the Gamma operator and Pochhammer notation, are covered in detail by Akmal et al. [21].

Table 1. FCN parametric architecture.

Layer	Type	No. of neurons	Total weights
Input	-	N	0
Hidden	Dense	288	N∗288 + 288 = 44,352
Dropout	Dropout	0	0
Output	Dense	2	288∗2 + 2 = 578

The Hahn coefficients derived from the previous equation are usually normalized using the coefficients presented in the subsequent equation.

$$H_{pq} = \Sigma^{G-1}_{j=0} \Sigma^{G-1}_{i=0}\ \delta_{pq} h^{a,b}{}_p\ (j, Q)\ h^{a,b}{}_q\ (i, Q), \quad m, n = 0, 1, 2, \ldots, Q - 1 \quad (12)$$

Classifiers

In this section deep classifiers such as FCN, RNN, CNN, LSTM, Bi-LSTM, and GRU have been utilized. In addition, a fine-tuned protein language ESM 2.0 model on benchmark dataset has been explained.

Fully connected network

In a fully connected neural network, there are multiple layers, starting with the input layer where the feature vector (X) is provided. After the input layer, each hidden layer contains neurons that are fully connected to every neuron in the subsequent layer. To make predictions, the input feature vector is passed through the network, and activations are computed for each neuron in the hidden layers. These activations are obtained by applying a weighted sum of the inputs feature values from the previous layer, along with bias terms. The result then goes through an activation function, like sigmoid or ReLU, to introduce non-linearity and enable the network to capture complex relationships between features [22].

This process is repeated for each hidden layer until we reach the final layer. The final layer generates the predictions for the given input, represented as Ŷ (Y-hat). To calculate the output Ŷ, we again perform a weighted sum of the outputs from the last hidden layer, along with a bias term, and then apply the activation function to obtain the predicted output. The final predictions (Ŷ) represent the network's estimation of the target output for the given input data. The formula for Ŷ can be written as:Ŷ=

$$\sigma\left(W(L) * A[L-1] + b[L]\right) \quad (13)$$

Where:

$\hat{\mathbf{Y}}$ is the predicted output vector. σ ()is the activation function applied elementwise to the vector. $\mathbf{W}[\mathbf{L}]$represents the weight matrix between the last hidden layer $(\mathbf{L}-1)$ and the output layer $\mathbf{L}$. $A[\mathbf{L}-1]$ denotes the activation vector from the last hidden layer.$\mathbf{b}[\mathbf{L}]$ is the bias vector for the output layer.

Figure 2 depicts the structural framework of the FCN Architecture, which was used to predict BBB peptides.

Table 1 demonstrates the FCN parametric architecture for BBB peptides classification.

Convolutional neural network

A word or word embedding sequence is the CNN's input. The input text is then transformed into a 2D matrix with a word embedding represented in each row [23]. The convolutional layer scans the

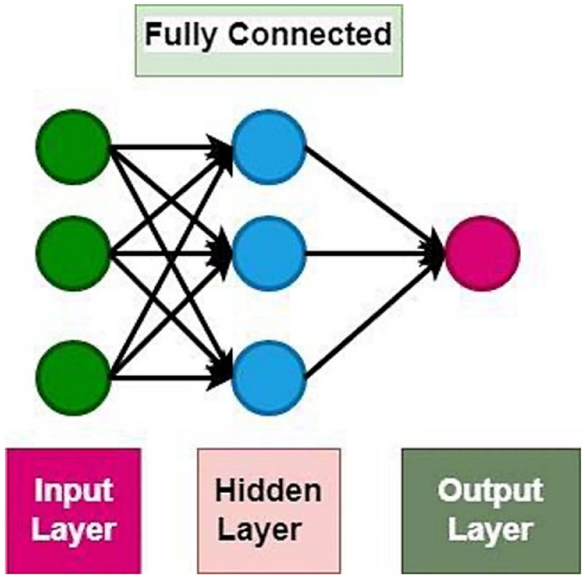


Figure 2. FCN graphical architecture.

input matrix using a collection of filters to extract pertinent n-grams (such as bi- or trigrams) of word embedding. The convolutional layer's feature maps are subsequently subjected to an activation function, such as ReLU. The final classification decision is often made by feeding the output of the convolutional layer through one or more fully linked layers. To reduce a loss function, like cross-entropy, the network is trained using gradient descent and back propagation [24].

The mathematical representation for outcome of CNN for text is presented as

$$y = f\left(Wx + b\right) \tag{14}$$

Where y represents the network's output, b symbolizes the bias term, f() exemplifies the activation function, W is the set of learnable filters, and x showcases the word embedding input matrix. Mathematically, the convolution operation itself is stated as follows:

$$y_i = f\left(\sum (j = 1 \text{ to } k)\ W_j * x_{(i+j-1)} + b\right) \tag{15}$$

Where x (i + j-1) denotes the input word embedding at position i + j-1, k represents the filter size, b represents the bias term, and y(i) exemplifies the output feature map at position i. W(j) is the weight at position j in the filter.

Figure 3 resembles the CNN architecture that has been employed in this study for identification of BBB peptides.

Table 2 shows the parameters architecture for CNN architecture used in this study.

Simple recurrent neural network

Recurrent neural networks (RNNs) address the limitation of FCNs in recognizing patterns at different sequence positions by sharing weights across time steps. This is achieved through a looping mechanism that allows the network to maintain a temporal state, effectively processing sequence vectors$x_1, x_2 \ldots . x_n$. The RNN updates its hidden state a_t at each time step t using the recurrence relation [25].

The recurrence relation is

$$a_t = f_a\left(\gamma_t - 1, x_t\right) \tag{16}$$

Where f is an activation function, a is a set of shared parameters, and x_t is the input at time stept. This structure enables RNNs to capture temporal dependencies and sequence patterns more effectively than FCNs.

Table 3 reflects the parametric architecture of Simple RNN for the BBB peptides identification.

Long short-term memory

The word embedding sequence serves as the LSTM's input. At each time step t, the LSTM analyzes the input sequence and produces a hidden state vector, h_t. Based on the current input word embedding x_t and the prior hidden state vector$h_{\{t-1\}}$, the hidden state vector h_t is updated. The current hidden state vector h_t is the output of the LSTM at each time step t. The last hidden state vector h_T is often passed through one or more fully connected layers to create the LSTM's final output [26]. To reduce a loss function, like cross-entropy, the network is trained using gradient descent and back propagation [23].

The output generated by an LSTM for a specific text input can be represented mathematically as follows:

$$h_t = f\left(Wx_t + Uh_{\{t-1\}}\right) + b\Big) \tag{17}$$

Where b is the bias term, f () is the activation function, such as the sigmoid or tanh function, W and U are the learnable weight matrices, and x_t is the input word embedding at time step t. To reach the ultimate classification conclusion, additional layers, such as a fully linked layer, can be used to process the output of the LSTM at each time step.

Figure 4 shows the employed LSTM architecture for prediction of BBB peptides.

Table 3 exemplifies the parametric values for LSTM architecture used in this study.

Gated recurrent unit

The GRU receives a series of word embedding as input. At each time step t, the GRU analyses the input sequence and produces a hidden state vectorh_t. Based on the current input word embedding x_t and the prior hidden state vectorh_{t-1}, the hidden state vector h_t is updated. The GRU features two gates that regulate how much of the prior hidden state should be retained and how much of the current input should be used. These gates are the update gate z_t and the reset gater_t. The current hidden state vector h_t is the output of the GRU at each time step t. The last hidden state vector h_T is often passed through one or more fully linked layers to produce the GRU's final output. To minimize a loss function, the network is trained via back propagation and gradient descent [18]. Table 5 shows the parametric architecture for GRU.

The mathematical expression for the output of a GRU given a specific text input is:

$$z_t = sigmoid\left(W_z * x_t + U_z * h_{\{t-1\}} + b_z\right) \tag{18}$$

$$r_t = sigmoid\left(W_r * x_t + U_r * h_{\{t-1\}} + b_r\right) \tag{19}$$

$$h\sim_t = \tanh\left(W_h * x_t + r_t * \left(U_h * h_{\{t-1\}} + b_h\right)\right) \tag{20}$$

$$h_t = (1 - z_t) * h_{\{t-1\}} + z_t * h\sim_t \tag{21}$$

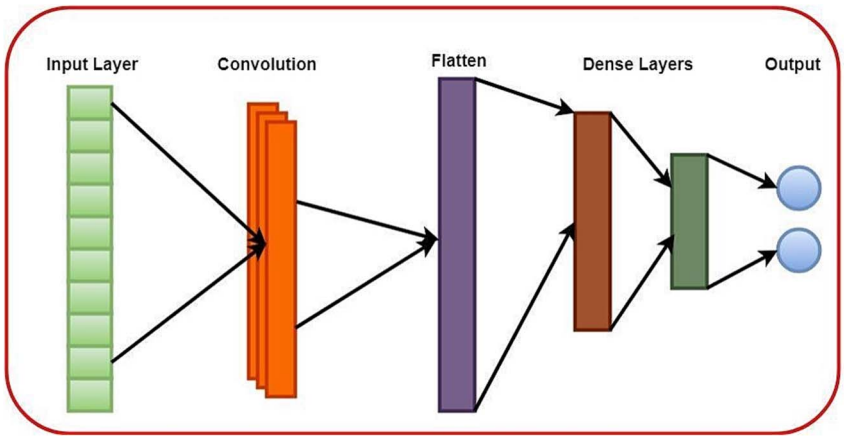


Figure 3. CNN graphical architecture.

Table 2. CNN parametric architecture.

Layer	Type	No. of neurons	Total weights
Input	-	N (number of input features)	0
Conv 1D	Conv 1D	464	4640
MaxPooling1D layer	MaxPooling1D layer	464	0
Conv1D layer	Conv1D layer	464	1 938 128
MaxPooling1D layer	MaxPooling1D layer	464	0
Flatten	Flatten	14,848	0
Dense	Dense	96	1 425 504
Output	Dense	2	194

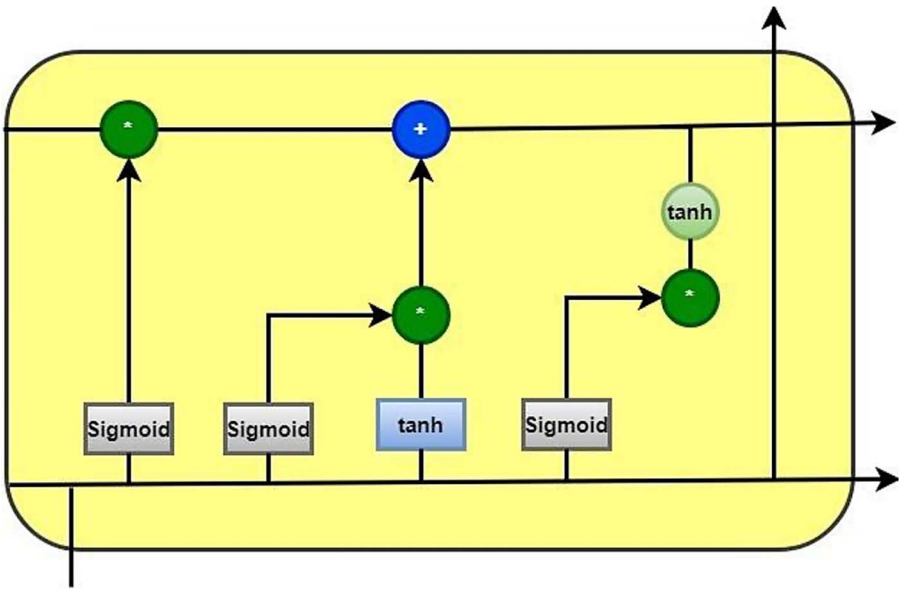


Figure 4. LSTM graphical architecture.

The learnable weight matrices are denoted byW_z, U_z, W_r, W_r, W_h, and W_h; the bias terms are B_z, B_r, and B_h; the sigmoid function is Sigmoid; and the hyperbolic tangent function is Tanh. The input word embedding at time step t is denoted byx_t. Each time step, the output of the GRU may be subjected to additional layers, such a fully connected layer, in order to determine the final classification decision.

Figure 5 demonstrates the GRU architecture used in this study for BBBPs identification.

Table 4 shows the parametric architecture for GRU architecture used in this study.

Bidirectional long short-term memory

A BidirectionalLSTM (Bi-LSTM) is an extension of the standard LSTM that improves model performance on sequence classification problems. Unlike a traditional LSTM that processes input sequences in one direction (typically from past to future),

Table 3. RNN parametric architecture.

Layer	Type	No. of neurons	Total weights
Input	-	N (153)	0
SimpleRNN	RNN	128	35,840
Dropout	Dropout	0	0
Dense	Dense	64	8256
Dropout	Dropout	0	0
Output	Dense	2	130

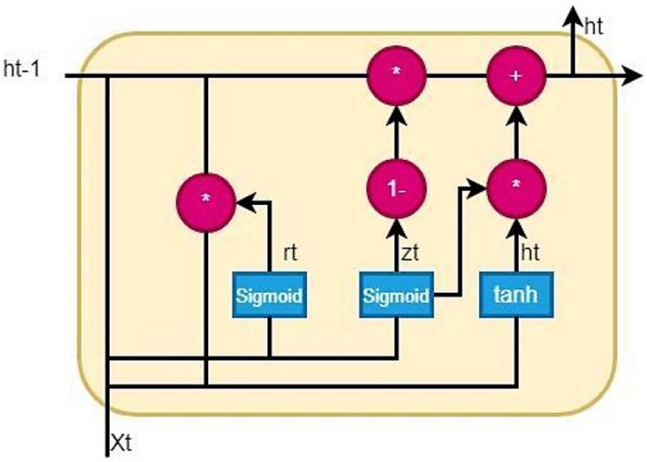


Figure 5. GRU graphical architecture.

Table 4. LSTM parametric architecture.

Layer	Type	No. of neurons	Total weights
Input	-	N (153)	0
LSTM	LSTM	128	144 384
Dropout	Dropout	0	0
Dense	Dense	64	8256
Dropout	Dropout	0	0
Output	Dense	2	130

Table 5. GRU parametric architecture.

Layer	Type	No. of neurons	Total weights
Input	-	N (153)	0
GRU	GRU	256	315 648
Dropout	Dropout	0	0
Dense	Dense	128	32 896
Dropout	Dropout	0	0
Output	Dense	2	258

a Bi-LSTM runs two LSTMs in parallel: one forward (from start to end) and one backward (from end to start). This bidirectional approach allows the model to capture information from both past and future contexts, leading to a richer understanding of the sequence data. The key difference from a regular LSTM is this dual processing, which enables Bi-LSTMs to achieve better performance on tasks where the context from both directions is important [27].

Table 6 exemplifies the Bi-LSTM parametric architecture for identification of BBB peptides.

Table 6. Bi-LSTM parametric architecture.

Layer	Type	No. of neurons	Total weights
Input	-	N (153)	0
Bidirectional	LSTM	128*2 = 256	295 936
Dropout	Dropout	0	0
Dense	Dense	64	16 448
Dropout	Dropout	0	0
Output	Dense	2	130

In the field of bioinformatics, supervised learning plays a key role in tasks like classifying genes and predicting protein structures. On the other hand, unsupervised learning helps researchers uncover hidden patterns in large sets of unlabeled genomic data. Transfer learning has also become invaluable, as it uses models trained on large biological datasets to tackle more specific tasks, especially when data is limited. Ensemble learning, which combines the strengths of multiple models, is particularly effective in areas like cancer classification, offering more reliable predictions. To enhance model performance, techniques like dropout (a form of regularization) and fine-tuning hyperparameters are commonly applied. For better interpretability, tools such as attention mechanisms and gradient-based visualizations allow researchers to understand how models make their decisions [28–31].

In this study, we focus on training deep neural networks, LSTM, Bi-LSTM, and CNN using supervised learning techniques. The training process is guided by gradient descent, ensuring the models learn effectively from the data.

Pre-trained Evolutionary Scale Modeling 2.0

ESM represents a significant leap forward in protein sequence analysis by harnessing the capabilities of protein language models to derive both structural and functional insights from amino acid sequences. Traditionally, protein structure prediction has heavily relied on techniques like multiple sequence alignments (MSAs) or structural templates. However, ESM-2, the most advanced model of its kind, breaks away from these conventional methods. By training exclusively on the distributional patterns found within raw protein sequences, it achieves impressive accuracy in predicting protein structures, as well as classifying proteins based on their functional attributes. This novel approach highlights how understanding the inherent patterns in sequence data can unlock deeper insights into protein biology.

One of the standout features of ESM-2 is its scale and architecture. As the largest protein language model to date, it comprises an immense 15 billion parameters, making it an exceptionally powerful tool in bioinformatics research. ESM-2 utilizes transformer-based architectures, which are adept at capturing complex relationships between sequences. By training on extensive protein datasets, the model can recognize intricate sequence relationships, which in turn are indicative of three-dimensional structures. This allows it to go beyond surface-level analysis, enabling more sophisticated predictions and insights.

A key application of ESM-2 is in structure prediction, facilitated by a specialized framework known as ESMFold. Leveraging ESM-2's deep sequence embeddings, ESMFold can perform end-to-end atomic-level structure predictions using just a single protein sequence as input [32]. This is a significant departure from other methods, such as AlphaFold, which often depend on MSAs for accurate structure prediction. By bypassing the need for homologous sequence alignments, ESMFold excels in scenarios where

sequence homology is low, making it particularly effective for characterizing orphan proteins those for which no known relatives exist in current databases.

Beyond structure prediction, ESM-2's capabilities extend to downstream tasks through fine-tuning. A scaled-down version of the model, comprising 35 million parameters, has been optimized for classification tasks. This involves predicting various attributes of proteins, such as their functional roles, cellular localization, and evolutionary families. Fine-tuning on specific tasks has demonstrated that scaling up the model leads to enhanced classification accuracy, underscoring that larger models can capture more nuanced and detailed biological information. As a result, this fine-tuned version is highly effective in annotating vast protein databases, particularly for sequences that are newly discovered and lack prior characterization.

Another critical advantage of ESM-2 is its ability to bridge the gap between sequence data and structural-functional insights. Unlike older models that rely on structural templates, ESM-2 leverages its understanding of sequence patterns to infer both structural and functional characteristics. This ability to predict a protein's structural conformation directly from its sequence data without relying on MSAs or templates marks a paradigm shift in the field. For researchers working with uncharacterized proteins, ESM-2 opens new avenues for hypothesis generation and exploration, providing functional insights that were previously inaccessible.

By moving away from dependency on traditional alignment-based methods, ESM-2 sets a new standard for how protein data can be analyzed and understood. It holds promise not only for structural biology but also for broader applications such as drug discovery, functional annotation, and even the exploration of evolutionary relationships among proteins. This model exemplifies the growing role of AI in expanding our understanding of the protein universe, pushing the boundaries of what can be achieved with purely data-driven approaches [33].

The ESM-2 model was fine-tuned with a learning rate of '2e-5', a batch size of 8, and 10 epochs. The number of labels is dynamically determined from the dataset. Regularization was applied with a weight decay of '0.01'. Evaluation and model saving occur after each epoch based on accuracy.

Evaluation metrics

To assess the performance of the employed classifiers, a range of evaluation metrics are utilized, including accuracy score, specificity, sensitivity, and Matthews's correlation coefficient. These metrics are used to evaluate the proposed model comprehensively. Out of all samples, the accuracy score indicates the total number of correct predictions from both classes [34, 35]. Specificity is a measure used in binary classification to evaluate the ability of a model to correctly identify true negatives from all negative samples. It quantifies the proportion of actual negatives that are correctly identified by the model as negatives [36]. The model's sensitivity indicates how well it can find positive instances [37]. As MCC takes into account both classes, it is a trustworthy statistic even in the case of unbalanced data [38]. An astute MCC score will be generated by the model if it is able to identify both the positive and negative samples. The formulas for each discussed metric are provided.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$Specificity = \frac{TN}{TN + FP} \tag{23}$$

Table 7. Results from the independent set.

Classifier	Accuracy	Sensitivity	Specificity	MCC
FCN	0.786	0.752	0.821	0.574
CNN	0.806	0.772	0.842	0.615
LSTM	0.756	0.728	0.789	0.513
GRU	0.75	0.727	0.779	0.502
RNN	0.776	0.772	0.779	0.55
Bi-LSTM	0.745	0.702	0.79	0.493
Fine-tuned ESM 2.0	0.843	0.824	0.861	0.686

$$Sensitivity = \frac{TP}{TP + FN} \tag{24}$$

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(FP + TP)\,(FN + TP)\,(FP + TN)\,(FN + TN)}} \tag{25}$$

A True Positive is a representation of the Peptides that the predictor correctly classified as being in the positive class. Peptides sequences that belong to the positive class but predicted as negative class by the predictor are referred to as false negatives. Conversely, False Positive denotes Negative samples that the predictor has tagged as Positive samples. A True Negative refers to samples that are correctly classified by the predictor as belongs to the negative class.

Results

To assess predictor robustness, three types of stringent tests were used: the self -consistency test, the independent set test and five-fold CV.

Self-consistency

The self-consistency test is a fundamental method used to assess the accuracy of a predictor. In this approach, the model was trained on the entire dataset without performing a train-test split to evaluate its performance. After training, all classifiers are tested to check if they successfully generated the model against the training dataset [39]. The self-consistency assessment demonstrates the model's coherence with the dataset it was trained on, as it undergoes training and evaluation using the same dataset [40]. The results for self-consistency have been attached to the supplementary material.

Independent testing

Independent testing is another way for evaluating a predictor's performance with unknown data. For this examination, the data is typically separated into two parts. The first section, which accounts for 77% of the total dataset, is used for training. This portion provides the model with input–output pairs to help it learn accurately. The remaining 23% is used to evaluate the predictor's accuracy. During this testing step, the predictor is just shown input features; the class name is not disclosed. Based on data that was unavailable during the training phase, the predictor produces predictions [41].

Table 7 shows that the Fine-tuned ESM 2.0 demonstrates exceptional performance, with accuracy score of 0.843. The obtained outcome from independent set test shows that the model functions effectively on data that was not seen during the training.

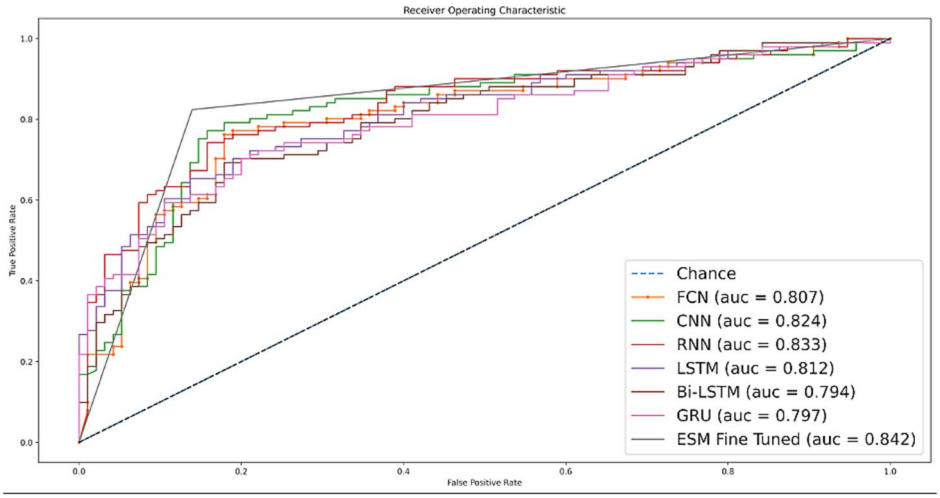


Figure 6. ROC curve for independent set test.

Table 8. Results using five-fold CV.

Classifier	Accuracy	Sensitivity	Specificity	MCC
FCN	0.758	0.722	0.79	0.514
CNN	0.747	0.725	0.771	0.5
RNN	0.749	0.723	0.772	0.5
LSTM	0.756	0.725	0.785	0.511
Bi-LSTM	0.753	0.72	0.783	0.504
GRU	0.746	0.69	0.8	0.493

Table 9. Comparison with benchmark studies.

Study	Accuracy	Sensitivity	Specificity	MCC
BBPpred [7]	66.67	65.66	67.68	33.34
B3Pred [8]	67.68	64.65	70.71	35.42
BBBPredict [9]	77.27	77.78	767.7	54.55
BBB-PEP-prediction [10]	82.4	83.1	91.1	66.3
BBB-PEP-ESM	84.3	86.1	82.4	68.6

Figure 6 visualizes the ROC for independent-set testing, the fine-tuned ESM 2.0 model outperforms other predictors.

Five-fold cross-validation

CV is a rigorous test that is utilized on every sample [42]. The process involves dividing the data into five equal folds. In each iteration, four of these folds are used for training, while one-fold is set aside for testing. This process is repeated until every fold has been used to generate predictions on every sample. The accuracy of each fold is calculated individually, and at the end, the average accuracy is computed. This method ensures that each sample of data is independently examined and trained, providing a thorough evaluation of the model's performance. Table 3 displays the outcomes for every classifier.

Among the utilized classifiers, the LSTM outperformed the others in five-fold CV test. It shows excellent accuracy and MCC score of 0.756 and 0.511, correspondingly. Table 8 shows how the results of the five-fold CV provide useful information into the predictor's effectiveness.

Figure 7 graphically represents the findings from the five-fold CV. The figure showcases that, RNN has outperformed other classifiers by attaining the AUC score of 0.805.

To graphically illustrate the distributions of multiple groups, the violin plot syndicates the qualities of boxplots and kernel density charts. Figure 8 demonstrates a violin chart illustrating the accuracy of five-fold CV findings.

Table 9 showcases comparative analysis for BBB peptides identification. The proposed approach BBB-PEP-ESM surpasses previous state-of-the-art benchmark studies. The suggested study's independent set testing yielded these results.

Table 10. List of peptides identification.

Sr	Peptide	Name	Status	Probability	Source
1	BBBP	JMV 2012	Identified	0.91	[12]
2	BBBP	TAPS	Identified	0.93	[12]
3	BBBP	PhrCACET1	Identified	0.91	[12]
4	BBBP	D3D3	Identified	0.92	[12]
5	BBBP	MVIIA-a	Identified	0.9	[11]
6	BBBP	Exendin-1	Identified	0.78	[11]
7	BBBP	Neuropeptide Y	Identified	0.86	[11]

Table 10 provides the list of BBBP's identification using proposed method. It shows the peptide sequences, trivial names, and their identification status.

Discussion

An in-silico technique has been used in this study for BBB peptides prediction. Initially, human-engineered features were used for peptides, with positional and composition variation features used to convert raw sequences into vector representation. The resultant vector was high-dimensional; thus, dimensionality was reduced by utilization of Raw, Hahn, and central-based statistical moments. This study gathered more information regarding the characteristics of peptides than earlier studies. State-of-the-art Deep learning architectures such as FCN, CNN, LSTM, and GRU were used to detect the BBB peptides. Keras Tuner was used to discover the ideal parameters for accuracy enhancement in FCN and CNN. The employed classifiers effectively discriminated between the positive and negative class by obtaining the high

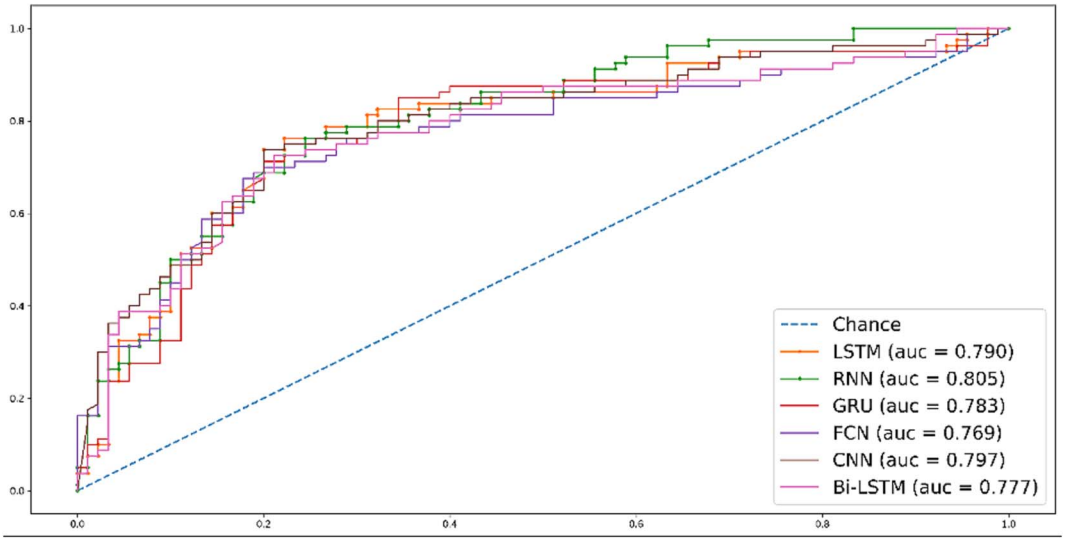


Figure 7. ROC graph using five-fold CV.

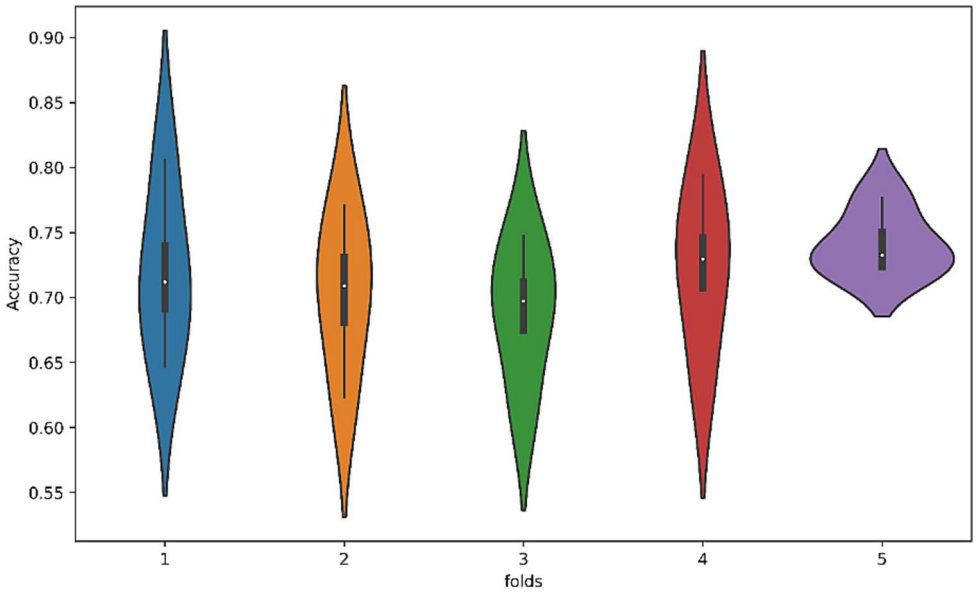


Figure 8. Violin plot using five-fold.

coefficients shown by feature space for BBB peptides prediction. To evaluate the predictor's performance, three testing methods were employed: self-consistency, independent set testing, and five-fold CV. Additionally, a pre-trained protein language model, ESM 2.0, was fine-tuned using a benchmark dataset to achieve better results than existing benchmark studies. The ESM 2.0 and CNN classifier consistently outperformed existing approaches in independent set test by obtaining accuracy score of 0.842 and 0.806 respectively. ESM 2.0, as a pre-trained model, has outperformed previous studies. Its capacity to handle varied Amino Acid patterns and record context-specific representations makes it resilient. The excellent results of ESM 2.0 originate from its deep architecture, large training data, and integration of cutting-edge transformer-based approaches, confirming its effectiveness in obtaining top-tier performance in performed studies. CNN attained ACC, SPE, SEN and MCC scores of 0.806, 0.842, 0.772, and 0.615 respectively in independent set test. In the independent set our previous study [9], showcases the highest accuracy score of 0.824. The fine-tuned model ESM 2.0 acquired an improved Accuracy score of 0.843. In addition, the table 10 highlights the sample BBBP's identified using BBB-PEP-ESM. When compared to other feature computation methodologies, the feature vector utilized in this study are vigorous and rigorous in capturing the qualities of sequences. Overall, the results, notably in independent set test, indicate that the predictor has a high level of generalization potential.

Boundary visualization

In the following section, we will guide you through the process of visualizing decision boundaries. Decision boundary is a line that divides sequences from individual class. Boundary visualization from individual deep classifier and raw data sequence visualization are two of the visualization approaches utilized in this study.

Figure 9 depicts the border visualization for each deep classifier, demonstrating the categorization of samples from both classes by establishing separate discrimination boundaries. Individual classifier created its own class discriminating space for both classes after passing through heterogeneous classifiers.

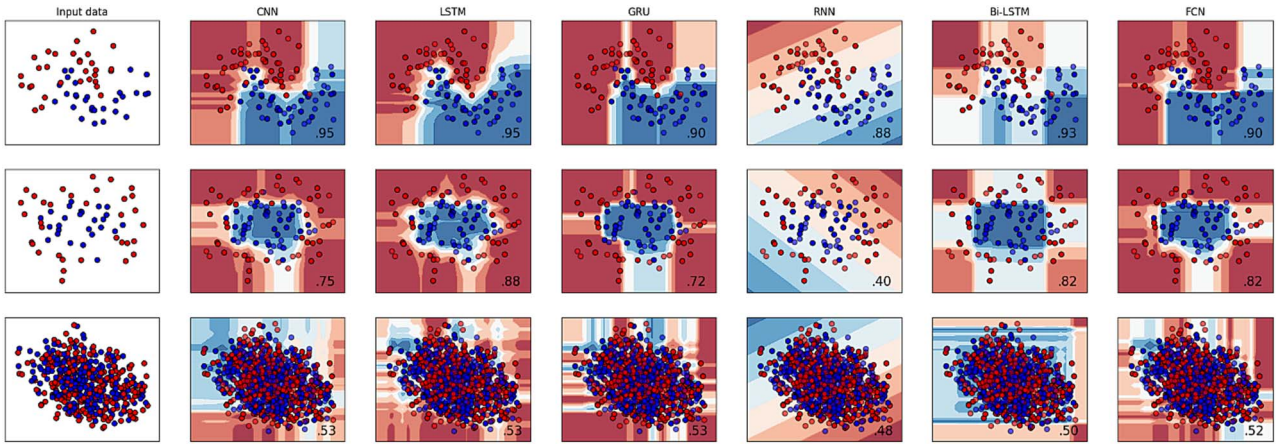


Figure 9. Boundary visualization for each classifier.

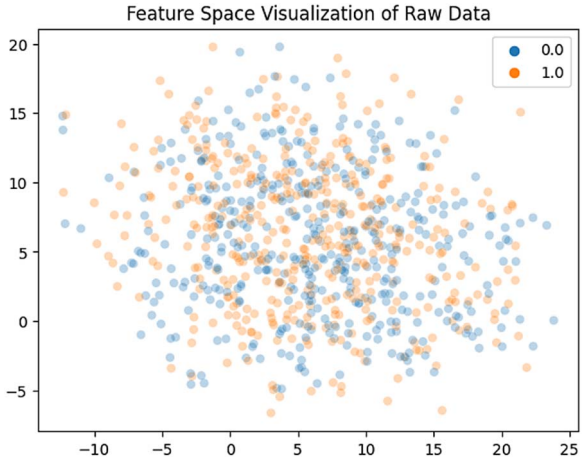


Figure 10. Feature space visualization.

The raw feature space representation, displayed in Fig. 10, illustrates the separateness of the raw data. The feature space presentation produces noteworthy results, demonstrating that the calculated features on BBB peptides have a substantial amount of discriminatory information. The presented study's classifiers effectively distinguish between two unique clusters in the data, each of which reflects a different class.

Conclusion

This research was carried out for computational identification of BBB peptides. The use of BBBPs can allow drugs to be transported into the brain, which creates new opportunities for the progress of therapies focused at treating CNS problems. This significance motivates the computational identification of these peptides. In this study, various feature computing techniques were used to extract hidden information inside sequences. For peptide sequences, feature vectors such as AAPIV, RAPPIV, PRIM, RPRIM, and FV have been computed. These feature computation techniques have provided important insights into protein characteristics. State-of-the-art Deep learning architectures such as FCN, CNN, RNN, LSTM, Bi-LSTM and GRU were employed to detect the BBB peptides. Keras Tuner was used to determine the best parameters to enhance accuracy. The utilized classifiers commendably discriminated between the two classes, for BBB peptides identification. To evaluate the performance, testing such as self-consistency, independent set testing, and five-fold CV have been employed. In addition, a pre-trained protein language model ESM 2.0 has been fine-tuned on benchmark dataset. In an independent set test, a fine-tuned ESM 2.0 has surpassed other classifiers with accuracy score of 0.843. ESM 2.0, as a pre-trained model, has proven its usefulness by outperforming the existing studies. Its ability to handle diverse Amino Acid patterns and capture context-dependent representations makes it robust. ESM 2.0's impressive results stem from its deep architecture, extensive training data, and the integration of state-of-the-art transformer-based methodologies, demonstrating its effectiveness in achieving top-tier performance in conducted study. In the independent set test, ESM 2.0 achieved accuracy, sensitivity, specificity and MCC scores of 0.843, 0.861, 0.824, and 0.686, respectively. This work investigates the identification of BBB peptides, and in the future, the BBB-PEP-Prediction can be employed for BBB peptide identification. It is worth mentioning, however, that this study has a disadvantage in that we have used the previous research dataset for experiment purposes. In the future work, we will increase the dataset to discover more peptides. Our research has made significant strides in accurately identifying BBB peptides. The model we developed, BBB-PEP-Prediction, outperforms previous approaches in terms of accuracy. However, one limitation is that we relied on existing datasets from earlier studies. This can limit the diversity of the data we used and might introduce biases, which could affect how well our model generalizes to new, unseen data. To address this, we plan to expand our dataset in the future by including a wider variety of peptide sequences. This will help improve the model's reliability and make it more effective for real-world applications.

Key Points

- BBBPs enhance drug delivery to the brain, overcoming the blood–brain barrier, which obstructs nearly 98% of small molecule drugs and almost all large molecule drugs.
- The study uses relative positions, reverse positions, and statistical moment-based features on benchmark dataset to improve BBB peptide identification.
- Six deep learning classifiers (FCN, CNN, RNN, LSTM, Bi-LSTM, and GRU) were utilized for effective BBB peptide classification.

- The Deep models have evaluated on self-consistency, independent set testing, and five-fold CV.
- The pre-trained protein language model ESM 2.0 was fine-tuned on a benchmark dataset and excelled in, independent set testing by surpassing existing benchmark studies in ACC, SPE, SEN, and MCC.

Acknowledgements

The researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

Author contributions

Conceptulization: A.N., Y.K. Methodology: T.A. Data Acquisition and Validation: F.A, A.N. Supervision: Y.K.

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Conflict of interest

The authors declare that they have no known financial or personal conflicts that could have influenced the research findings presented in this study.

Funding

The conducted research received no funding.

Data availability

The utilized benchmark dataset and code in this study is provided is uploaded on GitHub https://github.com/Ansar390/BBB_DEEP_ESM2.0 accessed on 2-2-2025.

References

1. Abbott NJ, Patabendige AA, Dolman DE. *et al.* Structure and function of the blood–brain barrier. *Neurobiol Dis* 2010;**37**:13–25. https://doi.org/10.1016/j.nbd.2009.07.030.
2. Tajes M, Ramos-Fernández E, Weng-Jiang X. *et al.* The blood–brain barrier: structure, function and therapeutic approaches to cross it. *Mol Membr Biol* 2014;**31**:152–67. https://doi.org/10.3109/09687688.2014.937468.
3. Abbott NJ, Rönnbäck L, Hansson E. Astrocyte–endothelial interactions at the blood–brain barrier. *Nat Rev Neurosci* 2006;**7**:41–53. https://doi.org/10.1038/nrn1824.
4. Friden PM, Walus LR, Watson P. *et al.* Blood–brain barrier penetration and in vivo activity of an NGF conjugate. *Science* 1993;**259**:373–7. https://doi.org/10.1126/science.8420006.
5. Chambers J. Delivery of therapeutics to the central nervous system. *Adv Drug Deliv Rev* 2012;**64**:589. https://doi.org/10.1016/j.addr.2012.02.009.
6. Pardridge WM. The blood–brain barrier: bottleneck in brain drug development. *NeuroRx* 2005;**2**:3–14. https://doi.org/10.1602/neurorx.2.1.3.
7. Dai R, Zhang W, Tang W. *et al.* BBPpred: sequence-based prediction of blood-brain barrier peptides with feature representation learning and logistic regression. *J Chem Inf Model* 2021;**61**:525–34. https://doi.org/10.1021/acs.jcim.0c01115.
8. Kumar V, Patiyal S, Dhall A. *et al.* B3pred: a random-forest-based method for predicting and designing blood–brain barrier penetrating peptides. *Pharmaceutics* 2021;**13**:1237. https://doi.org/10.3390/pharmaceutics13081237.
9. Chen X, Zhang Q, Li B. *et al.* BBPpredict: a web service for identifying blood-brain barrier penetrating peptides. *Front Genet* 2022;**13**:845747. https://doi.org/10.3389/fgene.2022.845747.
10. Naseem A, Alturise F, Alkhalifah T. *et al.* BBB-PEP-prediction: improved computational model for identification of blood–brain barrier peptides using blending position relative composition specific features and ensemble modeling. *J Chem* 2023;**15**:110. https://doi.org/10.1186/s13321-023-00773-1.
11. Kumar V, Patiyal S, Kumar R. *et al.* B3Pdb: an archive of blood–brain barrier-penetrating peptides. *Brain Struct Funct* 2021;**226**:2489–95. https://doi.org/10.1007/s00429-021-02341-5.
12. Van Dorpe S, Bronselaer A, Nielandt J. *et al.* Brainpeps: the blood–brain barrier peptide database. *Brain Struct Funct* 2012;**217**:687–718. https://doi.org/10.1007/s00429-011-0375-0.
13. Awais M, Hussain W, Khan YD. *et al.* iPhosH-PseAAC: identify phosphohistidine sites in proteins by blending statistical moments and position relative features according to the Chou's 5-step rule and general pseudo amino acid composition. *IEEE/ACM Trans Comput Biol Bioinform* 2019;**18**:596–610. https://doi.org/10.1109/TCBB.2019.2919025.
14. Ahmed S, Arif M, Kabir M. *et al.* PredAoDP: accurate identification of antioxidant proteins by fusing different descriptors based on evolutionary information with support vector machine. *Chemom Intel Lab Syst* 2022;**228**:104623. https://doi.org/10.1016/j.chemolab.2022.104623.
15. Khan YD, Alzahrani E, Alghamdi W. *et al.* Sequence-based identification of allergen proteins developed by integration of PseAAC and statistical moments via 5-step rule. *Curr Bioinforma* 2020;**15**:1046–55. https://doi.org/10.2174/1574893615999200424085947.
16. Ehsan A, Mahmood MK, Khan YD. *et al.* iHyd-PseAAC (EPSV): identifying hydroxylation sites in proteins by extracting enhanced position and sequence variant feature via chou's 5-step rule and general pseudo amino acid composition. *Curr Genomics* 2019;**20**:124–33. https://doi.org/10.2174/1389202920666190325162307.
17. Hussain W, Rasool N, Khan YD. A sequence-based predictor of Zika virus proteins developed by integration of PseAAC and statistical moments. *Comb Chem High Throughput Screen* 2020;**23**:797–804. https://doi.org/10.2174/1386207323666200428115449.
18. Khan YD, Khan NS, Naseer S. *et al.* iSUMOK-PseAAC: prediction of lysine sumoylation sites using statistical moments and Chou's PseAAC. *PeerJ* 2021;**9**:e11581. https://doi.org/10.7717/peerj.11581.
19. Butt AH, Khan YD. Prediction of S-sulfenylation sites using statistical moments based features via CHOU'S 5-step rule. *Int J Pept Res Ther* 2020;**26**:1291–301. https://doi.org/10.1007/s10989-019-09931-2.
20. Butt AH, Khan YD. CanLect-Pred: a cancer therapeutics tool for prediction of target cancerlectins using experiential annotated proteomic sequences. *IEEE Access* 2019;**8**:9520–31.
21. Akmal MA, Rasool N, Khan YD. Prediction of N-linked glycosylation sites using position relative features and statistical moments. *PloS One* 2017;**12**:e0181966. https://doi.org/10.1371/journal.pone.0181966.
22. Naseer S, Ali RF, Khan YD. *et al.* iGluK-deep: computational identification of lysine glutarylation sites using deep neural networks with general pseudo amino acid compositions. *J Biomol Struct Dyn* 2022;**40**:11691–704. https://doi.org/10.1080/07391102.2021.1962738.

23. Mehmood A, Farooq MS, Naseem A. *et al.* Threatening URDU language detection from tweets using machine learning. *Appl Sci* 2022;**12**:10342. https://doi.org/10.3390/app122010342.
24. Bodapati S, Bandarupally H, Shaw RN. *et al.* Comparison and analysis of RNN-LSTMs and CNNs for social reviews classification. *Adv Appl Data-Driven Comput* 2021;49–59. https://doi.org/10.1007/978-981-33-6919-1_4.
25. Naseer S, Fati SM, Muneer A. *et al.* iAceS-deep: sequence-based identification of acetyl serine sites in proteins using PseAAC and deep neural representations. *IEEE Access* 2022;**10**:12953–65. https://doi.org/10.1109/ACCESS.2022.3144226.
26. Naseer S, Ali RF, Fati SM. *et al.* Computational identification of 4-carboxyglutamate sites to supplement physiological studies using deep learning. *Sci Rep* 2022;**12**:128. https://doi.org/10.1038/s41598-021-03895-4.
27. Mughees N, Mohsin SA, Mughees A. *et al.* Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting. *Expert Syst Appl* 2021;**175**:114844. https://doi.org/10.1016/j.eswa.2021.114844.
28. Alharbi F, Vakanski A. Machine learning methods for cancer classification using gene expression data: a review. *Bioengineering* 2023; **10**:173. https://doi.org/10.3390/bioengineering10020173.
29. Prabhod KJ. The role of machine learning in genomic medicine: advancements in disease prediction and treatment. *J Deep Learn Genomic Data Anal* 2022;**2**:1–52.
30. Mathema VB, Sen P, Lamichhane S. *et al.* Deep learning facilitates multi-data type analysis and predictive biomarker discovery in cancer precision medicine. *Comput Struct Biotechnol J* 2023;**21**:1372–82. https://doi.org/10.1016/j.csbj.2023.01.043.
31. Li R, Li L, Xu Y. *et al.* Machine learning meets omics: applications and perspectives. *Brief Bioinform* 2022;**23**:bbab460. https://doi.org/10.1093/bib/bbab460.
32. Lin Z, Akin H, Rao R. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 2023;**379**:1123–30. https://doi.org/10.1126/science.ade2574.
33. Lin Z, Akin H, Rao R. *et al.* Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv* 2022, 2022;500902.
34. Ali Z, Alturise F, Alkhalifah T. *et al.* IGPred-HDnet: prediction of immunoglobulin proteins using graphical features and the hierarchal deep learning-based approach. *Comput Intell Neurosci* 2023;**2023**. https://doi.org/10.1155/2023/2465414.
35. Naseem A, Khan YD. An intelligent model for prediction of abiotic stress-responsive microRNAs in plants using statistical moments based features and ensemble approaches. *Methods* 2024;**228**:65–79. https://doi.org/10.1016/j.ymeth.2024.05.008.
36. Barukab O, Khan YD, Khan SA. *et al.* DNAPred_Prot: identification of DNA-binding proteins using composition- and position-based features. *Appl Bionics Biomech* 2022;**2022**:1–17. https://doi.org/10.1155/2022/5483115.
37. Alzahrani E, Alghamdi W, Ullah MZ. *et al.* Identification of stress response proteins through fusion of machine learning models and statistical paradigms. *Sci Rep* 2021;**11**:21767. https://doi.org/10.1038/s41598-021-99083-5.
38. Almagrabi AO, Khan YD, Khan SA. iPhosD-PseAAC: identification of phosphoaspartate sites in proteins using statistical moments and PseAAC. *Biocell* 2021;**45**:1287–98. https://doi.org/10.32604/biocell.2021.013770.
39. Amanat S, Ashraf A, Hussain W. *et al.* Identification of lysine carboxylation sites in proteins by integrating statistical moments and position relative features via general PseAAC. *Curr Bioinforma* 2020;**15**:396–407. https://doi.org/10.2174/1574893614666190723114923.
40. Barukab O, Khan YD, Khan SA. *et al.* iSulfoTyr-PseAAC: identify tyrosine sulfation sites by incorporating statistical moments via Chou's 5-steps rule and pseudo components. *Curr Genomics* 2019;**20**:306–20. https://doi.org/10.2174/1389202920666190819091609.
41. Alghamdi W, Alzahrani E, Ullah MZ. *et al.* 4mC-RF: improving the prediction of 4mC sites using composition and position relative features and statistical moment. *Anal Biochem* 2021;**633**:114385. https://doi.org/10.1016/j.ab.2021.114385.
42. Khan YD, Amin N, Hussain W. *et al.* iProtease-PseAAC (2L): a two-layer predictor for identifying proteases and their types using Chou's 5-step-rule and general PseAAC. *Anal Biochem* 2020;**588**:113477. https://doi.org/10.1016/j.ab.2019.113477.