BMC Bioinformatics

RESEARCH

Open Access

# Deep-ProBind: binding protein prediction with transformer-based deep learning model

Salman Khan[1], Sumaiya Noor[2], Hamid Hussain Awan[3], Shehryar Iqbal[4], Salman A. AlQahtani[5], Naqqash Dilshad[6] and Nijad Ahmad[7*]

*Correspondence:
Nijad@khurasan.edu.af

[1] Department of Computer Science, Abdul Wali Khan University Mardan, Mardan, KPK, Pakistan
[2] Business and Management Sciences Department, Purdue University, West Lafayette, IN, USA
[3] Department of Computer Science, Rawalpindi Women University, Rawalpindi 46300, Punjab, Pakistan
[4] School of Physics, Engineering and Computer Science, University of Hertfordshire, Hatfield, UK
[5] New Emerging Technologies and 5g Network and Beyond Research Chair, Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[6] Department of Computer Science & Engineering, Sejong University, Seoul 05006, South Korea
[7] Department of Computer Science, Khurasan University, Jalalabad, Afghanistan

## Abstract

Binding proteins play a crucial role in biological systems by selectively interacting with specific molecules, such as DNA, RNA, or peptides, to regulate various cellular processes. Their ability to recognize and bind target molecules with high specificity makes them essential for signal transduction, transport, and enzymatic activity. Traditional experimental methods for identifying protein-binding peptides are costly and time-consuming. Current sequence-based approaches often struggle with accuracy, focusing too narrowly on proximal sequence features and ignoring structural data. This study presents Deep-ProBind, a powerful prediction model designed to classify protein binding sites by integrating sequence and structural information. The proposed model employs a transformer and evolutionary-based attention mechanism, i.e., Bidirectional Encoder Representations from Transformers (BERT) and Pseudo position specific scoring matrix -Discrete Wavelet Transform (PsePSSM -DWT) approach to encode peptides. The SHapley Additive exPlanations (SHAP) algorithm selects the optimal hybrid features, and a Deep Neural Network (DNN) is then used as the classification algorithm to predict protein-binding peptides. The performance of the proposed model was evaluated in comparison with traditional Machine Learning (ML) algorithms and existing models. Experimental results demonstrate that Deep-ProBind achieved 92.67% accuracy with tenfold cross-validation on benchmark datasets and 93.62% accuracy on independent samples. The Deep-ProBind outperforms existing models by 3.57% on training data and 1.52% on independent tests. These results demonstrate Deep-ProBind's reliability and effectiveness, making it a valuable tool for researchers and a potential resource in pharmacological studies, where peptide binding plays a critical role in therapeutic development.

**Keywords:** Binding proteins, Deep learning, Transformer, Shap, Bert, PsePSSM

## Introduction

Peptides, short chains of amino acids typically under 50 residues, play vital roles in various cellular functions. Many function as hormones (e.g., insulin and oxytocin) [1], regulate cell signaling (e.g., signal peptides and proline-rich peptides), or contribute to defense mechanisms (e.g., antibiotics and antimicrobial peptides) [2]. Their diverse biological activities underscore the potential of peptide design for applications such as

modulating cell signaling, developing novel antibiotics, and targeting therapeutic factors like antibiotic resistance and programmed cell death receptors [3]. As biological catalysts, enzymes regulate numerous biochemical processes, and discovering ligands to modulate their activity is critical for advancing disease diagnosis and therapy [4, 5]. Peptides are emerging as promising molecules for enzyme modulation due to their chemical diversity, biocompatibility, and well-established peptide library synthesis methods [6, 7]. Peptides can influence enzyme activity, modulate receptor responses, and facilitate transmembrane transport, offering potential for therapeutic interventions [8]. Binding proteins, which selectively interact with peptides and other biomolecules, are fundamental to many biological processes, including enzymatic regulation, molecular transport, and immune system function. Their ability to recognize and bind specific targets is essential for modulating biochemical pathways and facilitating signal transduction. Understanding these interactions is key for designing peptides with enhanced binding properties, which can further optimize enzyme modulation and other therapeutic applications. However, challenges such as peptide size, low binding affinity, and conformational flexibility complicate the identification and characterization of binding proteins, highlighting the need for efficient prediction models [9, 10]. Consequently, developing novel, rapid, and precise computing tools is vital, a process revolutionized by recent developments in Artificial Intelligence (AI) and Machine Learning (ML).

ML can learn to predict these interactions better when trained on datasets of molecular libraries comprising ligand structures with their binding affinities, greatly accelerating chemical discovery [11, 12]. Numerous computational models have recently been developed to predict various protein-binding peptides. In predicting protein-binding peptides, computational methods generally fall into two broad categories, i.e., structure-based approaches [13] and sequence-based [14]. Structure-based methods rely on the three-dimensional structure of peptides and proteins to analyze molecular interactions, docking simulations, and binding free energy calculations. For instance, GraphPPIS [15] (Yuan et al., 2021), GraphBind [16] (Xia et al., 2021), PepNN-Struct [17] (Abdin et al., 2020), MaSIF-site [18] (Gainza et al., 2020), and SPPIDER [19] (Porollo and Meller, 2007). On the other hand, Sequence-based methods utilize peptide sequences and extract features such as amino acid composition, physicochemical properties, and evolutionary information to predict binding potential. These models often employ machine learning techniques trained on labeled datasets to identify patterns associated with binding affinity, i.e., ProNA2020 [20] and pepBCL [21]. Consequently, structure-based approaches rely on known tertiary structures, limiting their applicability to proteins with experimentally determined structures, which is time-consuming and complex. In contrast, sequence-based methods harness the power of Machine Learning (ML) to identify meaningful patterns within peptide sequences, enabling accurate predictions of binding affinities and interactions. These sequence-based approaches provide high-speed predictions with minimal computational cost, making them more efficient for large-scale analyses.

For instance, Romero-Molina et al. [22] highlight the challenges of virtual screening for protein–and protein-peptide interactions, emphasizing the complexity of predicting peptide-binding affinities for drug design. The proposed model, PPI-Affinity, uses support vector machine (SVM) models to predict binding affinities and rank mutants,

demonstrating its effectiveness on several benchmark datasets. Chandra et al. [23] proposed PepCNN, a deep learning-based model that integrates structural and sequence-based information. The proposed PepCNN enhances prediction accuracy by integrating half-sphere exposure, position-specific scoring matrices, and pre-trained protein language model embeddings. Recently, Azim et al. [24] emphasized the potential of peptides for modulating enzyme activity and proposed PepBind-SVM, an ML model for predicting protein-binding peptides based on sequential and physicochemical features. The proposed model achieves an average accuracy of 89.10% and 92.1% on training and independent testing using a new benchmark dataset. These ML algorithms have been explored to improve accuracy, but single-layer models struggle with nonlinear datasets. Further improvements can be achieved by incorporating transformer and evolutionary-based attention mechanisms with deep learning algorithms, highlighting the need for deep learning models that provide accuracy and interpretability [25].

This study addresses two key challenges in protein binding site prediction: the need for a reliable large-scale peptide-binding protein dataset and the development of a novel deep learning model. To address these issues, we introduce Deep-ProBind, a novel and robust prediction framework designed to accurately identify protein binding sites by leveraging protein sequence and structural information. Deep-ProBind encodes peptides using a transformer-based attention mechanism, specifically BERT, and generates evolutionarily informed features through a PsePSSM-DWT (Position-Specific Scoring Matrix with Discrete Wavelet Transform) approach. The model combines Word embeddings with evolutionary descriptors into a fused vector and optimally selects features using the SHAP (Shapley Additive Explanations) algorithm, which enhances interpretability and performance. Classification is then performed by a DNN, which enables the model to learn complex patterns within the data. Deep-ProBind demonstrated exceptional performance, achieving 92.67% accuracy on benchmark datasets through tenfold cross-validation and 93.62% accuracy on independent samples. It outperformed existing models by 3.57% on training data and 1.52% on independent tests, underscoring its reliability and superiority. These results position Deep-ProBind as a powerful tool for researchers, offering a dependable approach for peptide-binding site prediction. Its effectiveness also makes it a valuable resource in pharmaceutical research, where accurate peptide binding is crucial for the early stages of therapeutic development.

## Framework model

This section presents the design of the proposed model, as seen in Fig. 1, and a thorough description of every part of the model is presented to provide a complete understanding.

### Benchmark dataset

In bioinformatics and deep learning, selecting suitable training samples is crucial for constructing an effective predictive method. The selection of a benchmark dataset substantially influences the efficacy of a computational model. This work uses a benchmark dataset from [24] to verify our computational model for the training and assessment of the proposed approach. Initially, sample sequences were extracted for the training dataset, as depicted in Eq. 1.

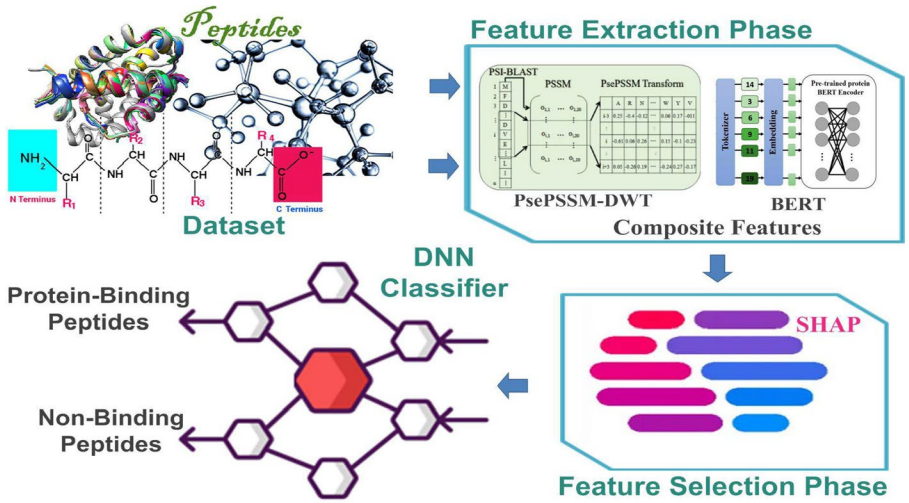


**Fig. 1** Framework of the proposed model

$$P = P^{+}UP^{-} \tag{1}$$

where P represents both protein-binding, i.e., $P^{+}$ and non-binding peptides, i.e.,$P^{-}$. We selected peptides with a binding affinity score above 0.8 (out of 1) as positive samples and those below 0.2 as negative samples. This approach made our dataset more explicit by having well-defined positive and negative examples. Peptides with scores between 0.2 and 0.8 were more complex to classify accurately, so we discarded them to ensure our samples were reliable. The goal of this study is to identify protein-binding and non-binding peptides accurately. After experimenting, we created a balanced training dataset with 1600 samples (800 positive and 800 negative) to avoid bias. For testing, we used an unbalanced set of 1000 samples to reflect better real-world conditions, i.e., 200 positive and 800 negative examples. Non-binding peptides are more common than positive ones; furthermore, in the independent dataset, no sequences from the training data were replicated to guarantee the generalization of the training model.

### Feature encoding schemes

In this section, we implement feature encoding schemes, as most predictive machine learning models require numerical data, which complicates peptide sequence representation. Feature extraction aids in this process, but choosing the right features is essential for model accuracy. The encoding must capture the sequence's structure and key characteristics.

#### *Position-specific scoring matrix (PSSM)*

The PSSM is the most geriatric and successful sequence alignment used to contrive the distant protein relatives by aligning multiple sequences. The first method, introduced by Gribskov et al. [26], captures the residue-level sequence-based similarities and structural characteristics using profile-based PSSM. In recent years, PSSM has proven to be an efficient predictor for several protein types like antifreeze proteins, RNA-binding proteins, hormone-binding proteins, DNA and interaction or Snare proteins, and various

membrane proteins [27]. PSSM for a protein sequence is computed using PSI-BLAST, which encodes the structural and evolutionary information in root symbols for residues of biological sequences. These categorical transitions are represented by the amino acid substitution scores offered by the matrix, compounding residue transformation probabilities over different residues (A to V, A to W). It reveals residues under evolutionary constraint, as positive scores (i.e., those with a higher score) tend to occur frequently and be evolutionarily conserved substitutions, and negative scores correspond to rare or unlikely substitutions. The PSSM matrix that we denote by M can be mathematically expressed as:

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,20} \\ \vdots & \vdots & \vdots & \vdots \\ h_{L,1} & h_{L,2} & \dots & h_{L,20} \end{bmatrix}_{Lx20} \tag{2}$$

where $h_{i,j}$ denoted the residue frequency of a peptide sample's $i^{th}$ and $j^{th}$ amino acid. L indicates the length of a given peptide sequence. Rows in matrix 'H' show the corresponding residues of the amino acids in a peptide sample, and columns in the matrix denote the twenty amino acids that are mutated. Where $h_{i,j}$ denoted the residue frequency of a peptide sample's $i^{th}$ and $j^{th}$ amino acid. L denotes the length of a specified peptide sequence. The rows of matrix 'H' represent the matching residues of amino acids in a peptide sample, whereas the columns indicate the twenty altered amino acids. The sigmoid function below normalizes the PSSM values $(X)_0 \rightarrow 1$.

$$PSSM(H) = \frac{1}{1 + e^{-x}} \tag{3}$$

This work employs protein pseudo and DWT methodologies on the PSSM matrix to get highly discriminative evolutionary descriptors.

#### *Pseudo-PSSM*

PSSM offers evolutionary insights, whereas variable-length protein sequences constrain machine-learning techniques such as SVM, RF, and KNN. Furthermore, PSSM neglects order information and association elements. PsePSSM addresses these issues by integrating sequence-order information to compute residue frequencies. PsePSSM applications include bioinformatics, proteomics, DNA-binding protein systems, and structural predictions for non-protein classes [28]. In this investigation, given a protein sequence of length L, PsePSSM has dimensions L*20, generated using the PSI-BLAST program to query the Swiss-Prot database [29]. Consequently, Pseudo-PSSM (PsePSSM) produces a consistent vector length from various peptide samples. PsePSSM computes the average score for each amino acid in the PSSM matrix by assessing the correlation between "d" residues. The PsePSSM vector for a peptide sample may be expressed as:

$$P_{psePSSM} = \left[\overline{P_1}, \overline{P_2}, \dots, \overline{P_{20}}, \emptyset_1^{\varepsilon}, \emptyset_2^{\varepsilon} \dots, \emptyset_{20}^{\varepsilon} \dots \emptyset_1^{lag}, \emptyset_2^{lag}, \dots \emptyset_{20}^{lag}\right]^T \tag{4}$$

where, $\overline{P_q} = \sum_{i=1}^{L} \frac{S_{p,q}}{L} (q = 1, 2 \dots 20)$, $\overline{P_q}$ denotes the mean score of all amino acid residues. Which are mutated to q amino acid in peptide sample 'S'

$$\emptyset_q^{\tau} = \frac{1}{L-\varepsilon}\sum_{i=1}^{L-\tau}\left[P_{i,q} - P_{(i+\tau),q}\right]^2,\ (q = 1, 2, \ldots, 20;\ \tau < L\ and\ \varepsilon \neq L) \tag{5}$$

where $\emptyset_q^{\tau}$ is the sequence ordering details of the peptide sample, q represents the amino acid, and $\tau$ is the contiguous distance.

#### *Discrete wavelet transform (DWT)*

DWT was introduced by Nanni et al. [30] to represent biological samples' frequency and residual information. DWT is an efficient signal compression and denoising method that disaggregates the amino acid sequence into many levels to uncover its latent features. DWT employs many factors to depict the PSSM matrix of a peptide sample as a picture. The PSSM image matrix is further partitioned into many levels according to its numerical coefficients to extract dependable and distinctive characteristics from the peptide sequence, which is not readily achieved by sequential encoding [31]. Each level is divided into two sub-wavelets: detail coefficients and approximation coefficients. The detailed coefficients denote the high-frequency (HF) values, while the approximation coefficients signify the low-frequency (LF) values. Prior research indicates that low-frequency components have more informational value than high-frequency components. Consequently, to examine the low-frequency components that are challenging to extract, we divide the LF portion of level-1 into HF1 and LF1 components of the subsequent level to get the concealed informative characteristics unattainable in level-1. Consequently, the following level of decomposition of the approximation coefficients results in the accumulation of highly discriminative features, as seen in Fig. 1. Ultimately, all characteristics (HF, LF) of level I and level II (HF1 and LF1) are amalgamated to create a modified feature set.

$$S(p,q) = \frac{1}{\sqrt{p}}\int_0^j y(i)\omega\left(\frac{j-q}{p}\right)d_j \tag{6}$$

where S (p, q) denotes the transformation coefficients, y(j) is the input signal, w(j-q/p) is the wavelet function and p, q represents the scaling and translation variables. We restrict our decomposition to two layers to eliminate noisy and duplicate characteristics. Consequently, after applying two 2-level DWT transformations to the PsePSSM matrix, a unique encoding method termed PsePSSM-DWT is established.

#### *Bidirectional encoder representations from transformers (BERT)*

Transformer-based models like BERT have made substantial progress in NLP by capturing contextual and semantic relationships [32]. This study uses the BERT architecture to extract features from peptide sequences, treating amino acids as words. Specifically, we use ProtBERT-BFD, which combines protein-based BERT embeddings with the Big Fantastic Database (BFD) [33] for enhanced feature representation. The peptide sequences are tokenized into individual amino acids, with a unique "CLS" token added to represent aggregated features for prediction. Each sequence is padded to a fixed length of 200 using "PAD" tokens, and a "SEP" token is used to separate sequences. The tokenized sequences are transformed into 1024-dimensional feature vectors using global average pooling. These

extracted features are then fed into the input layer of deep learning models for prediction. The overall mechanism of the ProtBERT-BFD model is illustrated in Fig. 2.

### Hybrid features

We used ProtBERT-BFD to gather contextual data and semantic associations from the peptide sequences. Conversely, the PsePSSM-DWT is used to gather changed evolutionary attributes. To augment the discriminative capacity of the training features with superior predictive efficacy, we amalgamated the extracted vectors (220D of PsePSSM-DWT and 1024D of BERT) into a hybrid vector (1244D) by making the necessary adjustments to offset the separate vectors' weaknesses.

$$H_{BP} = \text{PsePSSM}_{\text{DWT}} \cup \text{BERT}_{\text{BFD}} \tag{7}$$

### SHAP features selection

Decoding the biological import of selected features in machine learning models is sometimes problematic since these algorithms are known as black boxes, and their internal workings are complex to understand [34]. Another critical idea in machine learning is the data shape; it involves aspects like the organization, size, and arrangement of datasets utilized in a classification or regression function. Some behavioral patterns are exhibited by a machine learning algorithm based on the shape of the data sets it consists of. It is beneficial during data partition, such as dividing data into training, testing datasets, data normalization, and feature selection. Data cleaning is crucial because when data is well structured, it can perform optimally, hence the basis for decision-making. Through cooperative game theory, SHapley Additive Explanations (SHAP) can provide a solution to explain the contributions of 'input features' present in a model. SHAP scores each feature, and this numeric value encodes how informative that feature is to resulting decisions. The approach computes the prediction variation when a particular characteristic is included or excluded and quantifies its effect on the model [35]. This incremental effect is mathematically formalized through Eq. 8, which points out how feature iii impacts the result when interacting with different components of features.

$$SHAP_i(x) = \emptyset_i = \sum_{s \subseteq N\{i\}} \frac{|S|(|N| - |S| - 1)}{|N|} [f(S \cup \{i\}) - f(S)] \tag{8}$$

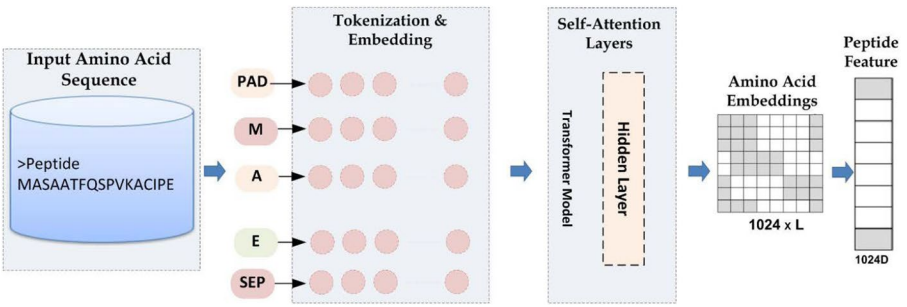


**Fig. 2** ProtBERT-BFD model using word embedding

where, $\phi_i$, denotes the SHAP value for the feature i. $N$, represents the set of all features. $S$, is a subset of features excluding feature i. $f(S)$ is the model's prediction given the features in S. $f(S \cup \{i\})$ is the model's prediction given the features in S and feature i.

In this study, we use BorutaSHAP-based wrapper feature selection to identify the most influential features from the extracted vector, as it evaluates the contribution of each feature to model performance [11, 36]. BorutaSHAP enhances the training process by highlighting the global importance of features and facilitating the selection of the optimal feature set. For our model, we selected the top 125 features from a hybrid feature vector with a total dimension of 1244. Figure 3 presents the summarized BorutaSHAP plots for the top 10 features, where each row represents a chosen feature. Red points indicate high-contributing features, while blue points signify those with lower contributions. The horizontal axis represents the SHAP values, with positive values indicating a prediction towards protein-binding peptides and negative values suggesting the non-binding peptides class.

### Deep architecture

Deep Neural Networks (DNNs) are a sub-classification of ML inspired by the structure and functionality of the human brain. DNN architecture involves an input layer, several hidden layers, and an output layer in between [37], as shown in Fig. 4.

The hidden layers are essential for the network to learn about features and patterns in data that it can't detect in the raw data. Whereas the number of hidden layers increases the predictive power to map complex patterns, it also increases the difficulty, computational costs, and over-fitting [38, 39]. Feature extraction is one of the most prominent advantages of DNNs as they do not need any feature engineering of the data since they can learn the features independently, even if the data is unlabeled or suffers from unstructured data. As pointed out in [40], this capability is realized

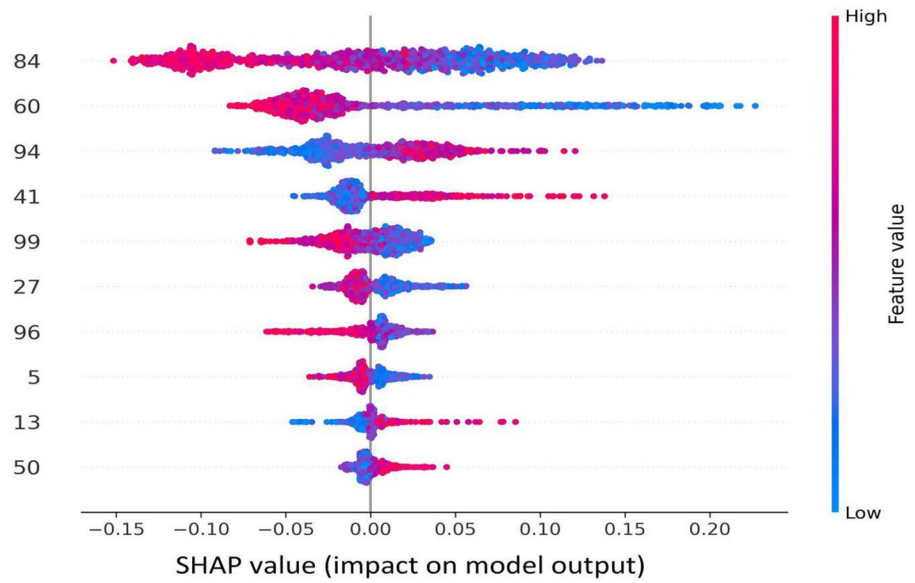


**Fig. 3** Feature selection via SHAP analysis

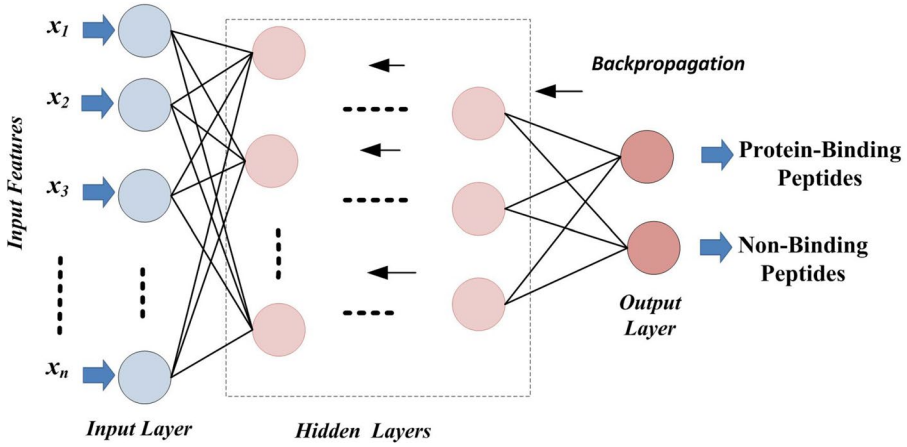


**Fig. 4** Neuron representation in DNN configuration

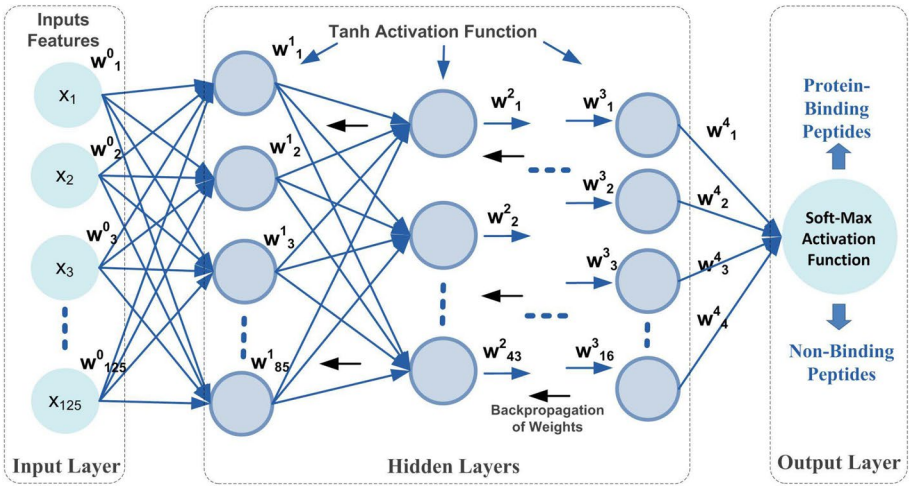


**Fig. 5** Proposed DNN model configuration

through standard learning methods. Experts have proved that DNNs are more effective in addressing complex classification problems than previous machine learning techniques [41, 42] because of their depth and flexibility. DNNs have been commonly used in many fields, such as bioengineering [43], speech recognition, image recognition [44], and natural language processing [36].

### *Model training*

Using a benchmark dataset, the DNN model is used to identify protein-binding peptides. The proposed DNN model comprises an input, output, and four hidden layers, as shown in Fig. 5. As with the previous novel architecture, each layer has multiple neurons, and the inputs and outputs correspond to the feature vectors shown in Eq. (9).

$$y_a = f\left(B_a + \sum_{b=1}^{m} x_b w_b^a\right) \tag{9}$$

where $y_a$ denote output at a layer, $B_a$ denote bias value, $w_b^a$ represent weight used at a layer $b$ by $a$ neuron, $x_b$ denote input feature, and $f$ denote a nonlinear activation Tanh function, which can be calculated using Eq. (10).

$$f(i) = \frac{e^i}{1 + e^i} \tag{10}$$

The weights stored at each neuron are set by the Xavier initialization method [45], ensuring that the variance is well-conserved and that practical learning is promoted across the layers. The proposed model learning technique is improved by using a backpropagation algorithm to change the weights iteratively, reducing errors between the output and target classes. The hyperbolic Tanh activation function [46] is used in both the input and hidden layers to incorporate nonlinearity into the developed model. This activation function enables the network to capture intricate patterns and the presence of relationships within data to decide whether a neuron should be activated because of the output generated. When measuring in the output layer, the activation function applied here is the softmax activation function. Since the probabilities of classifying the points or samples into an individual class, the values obtained are probabilities 0 (i.e., protein-binding peptides) and 1(i.e., non-binding peptides).

### Performance evaluation

Before deploying any machine learning model in a real-world setting, evaluating its performance is essential. While accuracy is necessary, it's not enough on its own. There are several other performance measures, such as Sensitivity (SN), Specificity (SP), Accuracy (ACC), Area Under the ROC Curve (AUC ROC), and Matthew's Correlation Coefficient (MCC). The best metric depends on the problem and how the model will be applied, as explained in [47]. As in other studies, we use these metrics to evaluate the proposed Deep Neural Network (DNN) performance. The performance metrics are calculated as follows:

$$ACC = \frac{T^+ + T^-}{T^+ + F^+ + T^- + F^-} \tag{11}$$

$$SP = \frac{T^-}{F^+ + T^-} \tag{12}$$

$$SN = \frac{T^+}{T^+ + F^-} \tag{13}$$

$$MCC = \frac{\left(T^- * T^+\right) - \left(F^- * F^+\right)}{\sqrt{\left(f^+ + T^+\right)\left(T^+ + F^-\right)\left(F^+ + T^-\right)\left(T^- + F^-\right)}} \tag{14}$$

where T+ symbolizes true positives, F+ symbolizes false positives, T− symbolizes true negatives, and F− false negatives, respectively.

## Discussion and experimental analysis

This section evaluates and discusses the proposed model's effectiveness in depth. Several validation tests, including the K-fold and independent tests, can be utilized to assess the overall performance of the machine learning training algorithm in bioinformatics. The K-fold cross-validation approach is a typical validation technique that uses evenly balanced findings. Consequently, a tenfold cross-validation test employing such benchmarking datasets was used to examine the overall accuracy of the suggested prescription in this work.

### System configuration

To experiment, we used the sixth-generation Intel Core i7 processor, an average desk work option that confidently performs its functions, such as data processing and basic computing tasks. SSD 256-GB, booting, reading, and writing speeds and application performance are much better than what HDD could provide. The system configuration also includes the 16 GB of RAM, which achieves a good level of multitasking. Typical Python 3 libraries such as Numpy and Scipy, common in data science workflows, were pre-installed onto the system for training and testing ML models. We also included Tensorflow and Keras [48] for building deep neural networks and Pandas and Matplotlib to do heavy work with data analysis, cleaning, and collating data for running machine learning models. This setup is well-suited for a data-centric individual or small members-focused team.

### Hyper parameters optimization

In this section, we intend to find the best values for the hyperparameters in the DNN model. We used a grid search algorithm [49] to assess DNN performance under different configurations. We noticed that the values of some parameters with the potential to improve DNN's performance were stochastic. A dropout rate of 0.25 and L2 regularization (0.001) are applied to prevent over-fitting. We included the following parameters in the grid search algorithm: activation function, Learning Rate (LR), and number of iterations. Based on the results, Table 1 shows a set of the best-obtained hyperparameter values.

We ran several experiments to evaluate how different activation functions and learning rates impact performance. The results, shown in Table 2, include tests using ReLU, Sigmoid, and Tanh as activation functions, with learning rates ranging from 0.08 to 0.5. According to Table 2, the DNN classifier achieved the highest accuracy, 92.67%, on the benchmark dataset when using Tanh as the activation function and a learning rate of 0.1.

Table 2 shows that a reduction in the learning rate results in an equal enhancement of the accuracy of the DNN model. However, decreasing the learning rate to less than 0. 1 did not produce much higher increases in accuracy. Therefore, we can also state that regarding the learning rate value, the DNN model reached the maximum accuracy, i.e., 0. 1, when using the Tanh activation function.

**Table 1** List of Optimal Hyper-parameters value of proposed DNN model

| List of parameters | Optimal values |
|---|---|
| Activation functions | Tanh and SoftMax |
| Regularization l2 | 0.001 |
| Number of hidden layers | 4 |
| Learning rates | 0.1 |
| Number of Neurons at hidden layers | 85-43-16-4 |
| Optimizer | SGD method |
| Updater | ADAGRAD function |
| Weight initialization function | XAVIER function |
| Seed | 12345L |
| Training Epoch | 50 |
| Dropout | 0.25 |
| Momentum | 0.9 |

**Table 2** Influence of LR and activation functions on the accuracy of the DNN model using a tenfold model

| LR | Tanh | Sigmoid | ReLU |
|---|---|---|---|
| 0.08 | 92.43 | 91.03 | 89.14 |
| 0.09 | 92.65 | 91.81 | 89.25 |
| 0.1 | 92.67 | 91.93 | 89.54 |
| 0.2 | 92.32 | 91.31 | 89.01 |
| 0.3 | 91.01 | 90.25 | 88.48 |
| 0.4 | 90.34 | 89.82 | 87.95 |
| 0.5 | 89.51 | 88.43 | 87.42 |

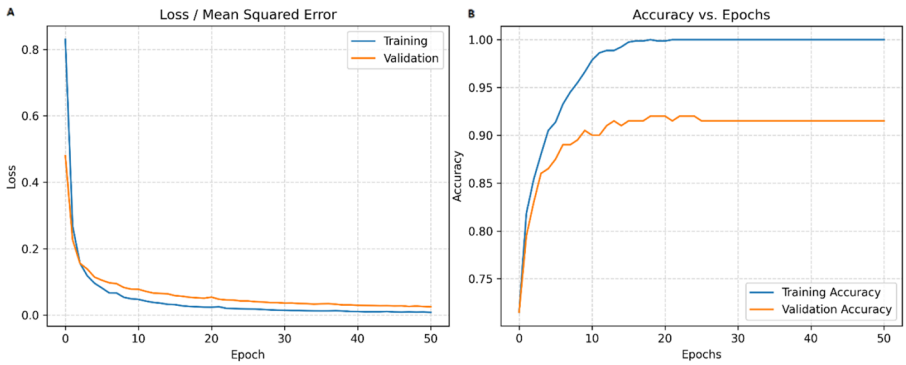


**Fig. 6** DNN model's performance by varying the number of training epochs. **A** epoch versus error loss **B** accuracy versus number epoch, using the tanh as activation functions

Next, we conducted numerous experiments to evaluate the DNN model's performance by varying the number of training epochs. The findings are illustrated in Fig. 6. Figure 6A shows the error loss versus the number of epochs using Tanh as the activation function. Figure 6A shows that the error rate consistently decreases as training epochs increase. For instance, the DNN model started with an error loss of 0.879 at the initial

epoch, steadily dropping to 0.001 by the 50th epoch. Similarly, Fig. 6B shows the accuracy results versus the number of epochs. From Fig. 6B, the training accuracy is 99.86%, nearly 100%. Similarly, the validation accuracy is also improved, at 92.67% by the 50th epoch. From these results, we can conclude that 50 epochs are optimal, as the error rates stabilize and accuracy is improved at these points. The optimal configuration derived from this analysis is summarized in Table 1.

#### Performance analysis of DNN

In this section, we evaluate the proposed model on different sequence formulation methods and the hybrid features using training and independent datasets, as summarized in Table 3. The sequence formulation methods include PsePSSM-DWT and BERT and the hybrid features with and without feature selection. For instance, on the training dataset, the PsePSSM-DWT method achieved an accuracy of 89.26% with an MCC of 0.795, while the BERT model performed slightly lower, with an accuracy of 87.17% and an MCC of 0.748. Moreover, the hybrid feature method outperformed the sequence formulation methods, achieving an accuracy of 90.28%, with an MCC of 0.812. The main reason is that hybrid feature methods combine the strengths of multiple feature sets, capturing more comprehensive and diverse information relevant to the prediction task.

In order to further improve the performance of the proposed model, the dimensionality of the hybrid features space was reduced using the feature selection method. Applying feature selection in hybrid models helps eliminate redundant or irrelevant features, enhancing the model's performance by focusing on the most informative features. As a result, the success rate of the proposed model was significantly improved, i.e., the average accuracy enhanced to 92.67%, Sensitivity of 93.41%, F1 score of 93.40%, Specificity of 91.82%, and an MCC of 0.853. These results demonstrate that incorporating feature selection into the hybrid model significantly improves classification performance across all metrics.

Furthermore, the performance of individual and hybrid features was evaluated on an independent dataset. The BERT model achieved an accuracy of 91.20% with an MCC of 0.824, while the PsePSSM-DWT method performed slightly better, attaining an accuracy of 91.49% and an MCC of 0.830, indicating predictive solid capabilities for both models. When comparing the hybrid feature method without feature selection, it achieved higher metrics across the board, with an accuracy of 92.72% and an MCC of 0.856, suggesting

**Table 3** Performance comparison using sequence formulation techniques and hybrid feature vector

| Methods | Dataset | ACC (%) | SN (%) | F1 (%) | SP (%) | MCC |
|---|---|---|---|---|---|---|
| BERT | Training | 87.17 | 88.17 | 87.36 | 86.18 | 0.748 |
| PsePSSM-DWT | | 89.26 | 90.42 | 89.04 | 88.10 | 0.795 |
| Hybrid feature (without feature selection) | | 90.28 | 91.14 | 90.36 | 89.35 | 0.812 |
| Hybrid features (with feature selection) | | 92.67 | 93.41 | 93.40 | 91.82 | 0.853 |
| BERT | Independent | 91.20 | 91.91 | 90.91 | 90.50 | 0.824 |
| PsePSSM-DWT | | 91.49 | 92.47 | 91.46 | 90.47 | 0.830 |
| Hybrid feature (without feature selection) | | 92.72 | 93.84 | 92.75 | 91.71 | 0.856 |
| Hybrid features (with feature selection) | | 93.62 | 94.36 | 94.35 | 92.82 | 0.872 |

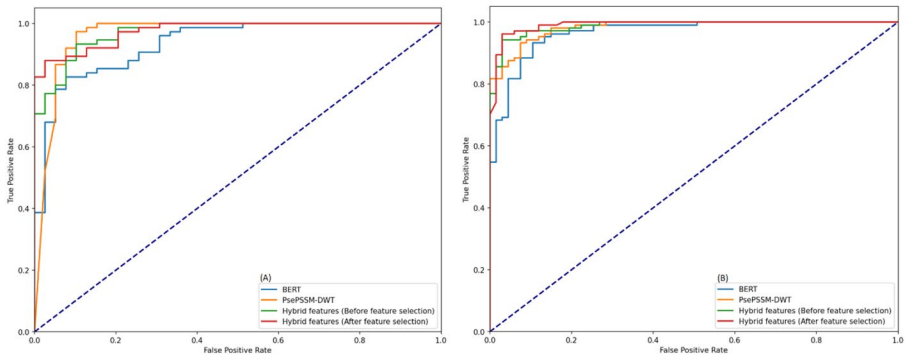


**Fig. 7** AUC performance comparison on training **A** and independent **B** dataset using 10-Fold Cross Validation

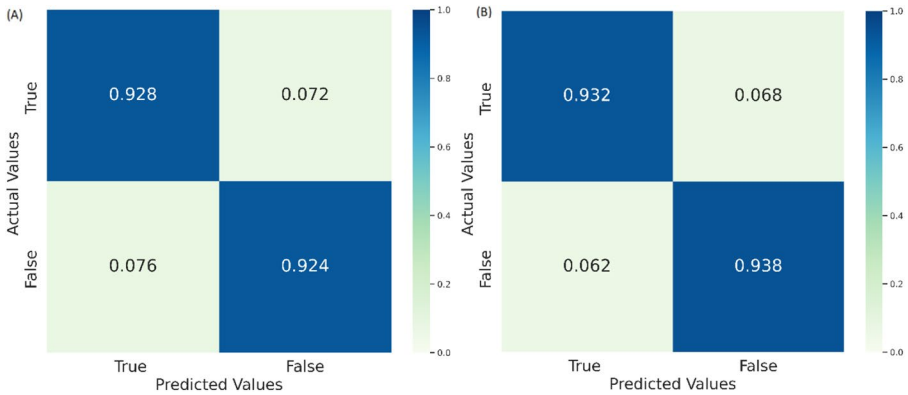


**Fig. 8** Confusion matrix of the proposed model on training **A** and independent **B** dataset

that combining features can enhance overall performance. The hybrid features with feature selection demonstrated the best performance, with an accuracy of 93.62%, Sensitivity of 94.36%, Specificity of 92.82%, an F1 score of 94.35%, and an MCC of 0.872. This indicates that incorporating feature selection improved the model's ability to discern relevant patterns and helped reduce noise from irrelevant or redundant features, leading to superior classification outcomes. Overall, the results illustrate that hybrid features, particularly when optimized through feature selection, significantly enhance predictive performance compared to individual methods.

The performance of Deep-ProBind was further examined using the AUC metric [44], a key indicator of binary classifier accuracy. A higher AUC score directly translates to better model performance. As illustrated in Fig. 7A, B Deep-ProBind delivered outstanding results, achieving an AUC of 0.941 on the training dataset (i.e., Fig. 7A) and 0.948 on the independent dataset (i.e., Fig. 7B) using tenfold cross-validation. These findings highlight the model's exceptional predictive capabilities, particularly when using selected features on the tenfold cross-validation method. Additionally, Fig. 8A, B presents a confusion matrix that delves deeper into the performance of the DNN classifier, showcasing its effectiveness in the prediction with selected features vector on the training dataset (i.e., Fig. 8A) and independent dataset (i.e., Fig. 8B).

### Performance comparison with different classifiers

In this section, the performance of the proposed model is examined by testing it with several well-known supervised machine learning algorithms on optimized hybrid features using training and independent datasets. The nature of the algorithms under consideration for this comparison are Random Forest (RF) [50], Support Vector Machine (SVM)[51], Logistic Regression (LR) [52], Naive Bayes (NB) [53], and K- Nearest Neighbor (KNN) [54]. Random Forests is another ensemble learning technique that builds several decision trees by utilizing different bootstrapping methods. The result from each tree decision is combined by applying voting to improve the classification performances. It can be used in almost all classification and regression problems. K-Nearest Neighbor is a non-parametrized learning algorithm widely used in image processing. It divides instances into classes depending on the distance from the neighbors, and due to the straightforward approach, it fits most of the problems. Support Vector Machines [55] are especially effective when dealing with linear and nonlinearly separable data; this algorithm searches for the best hyperplane to classify different classes effectively. This method is widely used, especially in bioinformatics, because of its effectiveness in working with large data sets. Naive Bayes, which derives from Bayes' Theorem, is a probabilistic classifier that analyzes features independently. It is particularly effective for text categorization, having small data sets, and working in high-dimensional spaces. Logistic Regression (LR) is a standard statistical method used for binary classification, where it calculates the probability of an outcome belonging to one of two categories. It employs the maximum likelihood estimation method to establish the relationship between a dependent variable with two possible outcomes and one or more independent variables. LR is widely valued for its simplicity, interpretability, and effectiveness in real-world applications, such as medical diagnosis and credit risk assessment. Table 4 shows the proposed model performance with commonly used learning algorithms on training and independent datasets.

Table 4 compares different models applied to the training dataset using various evaluation criteria. The presented models' performance can be analyzed based on accuracy: the Logistic Regression (LR) model reached 88.48% and an MCC of 0. 769. Respectively,

**Table 4** Performance comparison of different classifiers

| Methods | Dataset | ACC (%) | SN (%) | F1 (%) | SP (%) | MCC |
|---|---|---|---|---|---|---|
| LR | Training | 88.48 | 89.03 | 88.03 | 87.89 | 0.769 |
| NB | | 89.26 | 91.04 | 89.04 | 89.41 | 0.782 |
| RF | | 89.91 | 90.65 | 88.65 | 88.59 | 0.795 |
| KNN | | 90.25 | 91.36 | 90.36 | 89.28 | 0.812 |
| SVM | | 91.15 | 92.42 | 91.47 | 90.01 | 0.830 |
| Deep-ProBind | | 92.67 | 93.41 | 93.41 | 91.82 | 0.853 |
| LR | Independent | 91.61 | 92.22 | 92.00 | 91.06 | 0.832 |
| NB | | 92.01 | 92.86 | 92.85 | 91.22 | 0.840 |
| RF | | 92.15 | 92.57 | 92.71 | 91.77 | 0.843 |
| KNN | | 92.45 | 93.79 | 91.53 | 91.27 | 0.846 |
| SVM | | 92.77 | 94.06 | 91.17 | 91.56 | 0.851 |
| Deep-ProBind | | 93.62 | 94.36 | 94.90 | 92.82 | 0.872 |

**Table 5** Performance compared with existing predictors on training and independent datasets

| Method | Dataset | ACC (%) | SN (%) | SP (%) | MCC |
|---|---|---|---|---|---|
| Deep-ProBind | Training | 92.67 | 93.41 | 91.82 | 0.853 |
| PepBind-SVM [24] | | 89.10 | 85.40 | 92.90 | 0.784 |
| Deep-ProBind | Independent | 93.62 | 94.36 | 92.82 | 0.872 |
| PepBind-SVM [24] | | 92.10 | 86.00 | 93.60 | 0.765 |

the NB, RF, and KNN models had better performances, with a mean accuracy of 89.26%, 89.91%, and 90.25%. The MCC values of the three sets were 0.782, 0.795, and 0.812. The SVM model recorded the highest accuracy amongst the traditional models at 91.15% with an MCC of 0.830. However, the proposed model yielded the highest performance with an accuracy of up to 92.67% and an MCC of up to 0.853. Therefore, it demonstrates the higher efficiency of the proposed model compared to the other methods used in the analysis.

Moreover, Table 4 also presents the performance metrics for various models on the independent dataset. The SVM model had a maximum accuracy of 92.77% and an MCC of 0.851 compared to the other traditional machine learning techniques. The proposed model outperformed all others, with the highest accuracy of 93.62% and an MCC of 0.872. Similarly, in the other parameters, the f1 score is 94.90%, Sensitivity 94.36%, and Specificity 92.82%. The DNN model outperformed the SVM algorithm and other traditional ML because it uses a single processing layer. Traditional ML struggles with complex datasets that have high nonlinearity.

### Performance comparison with existing models

We evaluate the performance of the proposed Deep-ProBind model against the existing model on both training and independent datasets. Table 5 compares our proposed predictor, Deep-ProBind, against an existing predictor, PepBind-SVM [1], using training and independent datasets. From the Table, the proposed Deep-ProBind shows impressive accuracy, achieving 92.67% on the training dataset and 93.62% on the independent dataset, which surpasses PepBind-SVM's Accuracy of 89.10% and 92.10%, respectively. In terms of Sensitivity, our proposed Deep-ProBind also excels, with values of 93.41% in the training set and 94.36% in the independent set, indicating a solid ability to correctly identify positive instances compared to PepBind-SVM, which recorded 85.40% and 86.00% Sensitivity. These results illustrate that Deep-ProBind outperforms PepBind-SVM across all metrics, demonstrating its potential as a highly effective predictor for identifying protein-binding peptides.

## Conclusions

Protein-binding peptides are vital in several biological processes, rendering their precise prediction critical for the progression of drug discovery and therapeutic development. To further understand its biological origin, we proposed a unique deep learning-based model called Deep-ProBind, a computational model. The proposed model was designed to accurately predict protein-binding peptides by leveraging optimized hybrid features and utilizing tenfold cross-validation and independent datasets. The model effectively

addressed the over-fitting issue by optimizing hyper-parameters and demonstrated robust performance, achieving accuracies of 92.67% and 93.62% on the training and independent datasets, respectively. Furthermore, the model's average accuracy on training and independent test samples demonstrates its superiority over conventional machine learning techniques and current state-of-the-art approaches. The encouraging results of Deep-ProBind underscore its potential to substantially advance research in finding functional peptides, their relevance in diseases, especially in stress response and breast cancer, and their use in formulating treatment methods.

For future work, we plan to explore the integration of transfer learning to improve the model's adaptability across diverse datasets. Refining the model architecture through hyperparameter optimization and employing ensemble techniques could enhance performance and robustness. We aim to incorporate parallel programming approaches to address scalability and efficiency, ensuring faster and more resource-efficient processing. A key limitation of the current study is the reliance on a relatively small dataset, which may restrict the model's generalizability. We aim to overcome this limitation by incorporating more extensive and diverse datasets in future iterations.

**Acknowledgements**
This work was supported by Research Supporting Project Number (RSPD2025R585), King Saud University, Riyadh, Saudi Arabia.

**Author contributions**
All authors contributed equally. SK and SN wrote the main manuscript text. SAQ and NA debug the code, provide datasets, and HHA, SI and ND review the paper and grammar.

**Funding**
This research is not funded.

**Availability of data and materials**
The datasets used and/or analyzed during the current study are available on the GitHub link: https://github.com/salman-khan-mrd/Deep-ProBind.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare no competing interests.

Received: 3 November 2024 Accepted: 4 March 2025
Published online: 22 March 2025

**References**

1. GoulardCoderc de Lacam E, Roux B, Chipot C. Classifying protein-protein binding affinity with free-energy calculations and machine learning approaches. J Chem Inf Model. 2024;64(3):1081–91. https://doi.org/10.1021/acs.jcim.3c01586.
2. Mohaideen NSMH, Vaani S, Hemalatha S. Antimicrobial peptides. Curr Pharmacol Reports. 2023;9(6):433–54. https://doi.org/10.1007/s40495-023-00342-y.
3. Tallorin L, et al. Discovering de novo peptide substrates for enzymes using machine learning. Nat Commun. 2018;9(1):5253. https://doi.org/10.1038/s41467-018-07717-6.
4. Xiao X, Zou HL, Lin WZ. iMem-Seq: a multi-label learning classifier for predicting membrane proteins types. J Membr Biol. 2015;248(4):745–52. https://doi.org/10.1007/s00232-015-9787-8.
5. Bibi N, et al. Sequence-based intelligent model for identification of tumor T cell antigens using fusion features. IEEE Access. 2024;12:155040–51. https://doi.org/10.1109/ACCESS.2024.3481244.
6. Robinson PK. Enzymes: principles and biotechnological applications. Essays Biochem. 2015;59:1–41. https://doi.org/10.1042/bse0590001.

7. Radley E, Davidson J, Foster J, Obexer R, Bell EL, Green AP. Engineering enzymes for environmental sustainability. Angew Chemie Int Ed. 2023;62(52):e202309305. https://doi.org/10.1002/anie.202309305.
8. Efremov DG, Turkalj S, Laurenti L. Mechanisms of B cell receptor activation and responses to b cell receptor inhibitors in B cell malignancies. Cancers (Basel). 2020;12(6):1396. https://doi.org/10.3390/cancers12061396.
9. Khan S, et al. Sequence based model using deep neural network and hybrid features for identification of 5-hydroxymethylcytosine modification. Sci Rep. 2024;14(1):9116. https://doi.org/10.1038/s41598-024-59777-y.
10. Uddin I, et al. A hybrid residue based sequential encoding mechanism with XGBoost improved ensemble model for identifying 5-hydroxymethylcytosine modifications. Sci Rep. 2024;14(1):20819. https://doi.org/10.1038/s41598-024-71568-z.
11. Khan F, et al. Prediction of recombination spots using novel hybrid feature extraction method via deep learning approach. Front Genet. 2020;11:1052. https://doi.org/10.3389/fgene.2020.539227.
12. Inayat N, et al. iEnhancer-DHF: identification of enhancers and their strengths using optimize deep neural network with multiple features extraction methods. IEEE Access. 2021;9:40783–96. https://doi.org/10.1109/ACCESS.2021.3062291.
13. Wang DD, Wu W, Wang R. Structure-based, deep-learning models for protein-ligand binding affinity prediction. J Cheminform. 2024;16(1):2. https://doi.org/10.1186/s13321-023-00795-9.
14. Lee I, Nam H. Sequence-based prediction of protein binding regions and drug–target interactions. J Cheminform. 2022;14(1):5. https://doi.org/10.1186/s13321-022-00584-w.
15. Yuan Q, Chen J, Zhao H, Zhou Y, Yang Y. Structure-aware protein–protein interaction site prediction using deep graph convolutional network. Bioinformatics. 2021;38(1):125–32. https://doi.org/10.1093/bioinformatics/btab643.
16. Xia Y, Xia C-Q, Pan X, Shen H-B. GraphBind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues. Nucleic Acids Res. 2021;49(9):e51–e51. https://doi.org/10.1093/nar/gkab044.
17 Abdin O, Nim S, Wen H, Kim PM. PepNN: a deep attention model for the identification of peptide binding sites. Commun Biol. 2022;5(1):503. https://doi.org/10.1038/s42003-022-03445-2.
18. Gainza P, et al. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nat Methods. 2020;17(2):184–92. https://doi.org/10.1038/s41592-019-0666-6.
19. Langdon QK, Peris D, Kyle B, Hittinger CT. sppIDer: a species identification tool to investigate hybrid genomes with high-throughput sequencing. Mol Biol Evol. 2018;35(11):2835–49. https://doi.org/10.1093/molbev/msy166.
20. Qiu J, et al. ProNA2020 predicts protein–DNA, protein–RNA, and protein–protein binding proteins and residues from sequence. J Mol Biol. 2020;432(7):2428–43. https://doi.org/10.1016/j.jmb.2020.02.026.
21. Hu J, et al. Protein-peptide binding residue prediction based on protein language models and cross-attention mechanism. Anal Biochem. 2024;694:115637. https://doi.org/10.1016/j.ab.2024.115637.
22. Romero-Molina S, et al. PPI-Affinity: a web tool for the prediction and optimization of protein-peptide and protein-protein binding affinity. J Proteome Res. 2022;21(8):1829–41. https://doi.org/10.1021/acs.jproteome.2c00020.
23. Chandra A, Sharma A, Dehzangi I, Tsunoda T, Sattar A. PepCNN deep learning tool for predicting peptide binding residues in proteins using sequence, structural, and language model features. Sci Rep. 2023;13(1):20882. https://doi.org/10.1038/s41598-023-47624-5.
24. Azim SM, Balasubramanyam A, Islam SR, Fu J, Dehzangi I. Explainable machine learning model to accurately predict protein-binding peptides. Algorithms. 2024;17(9):409. https://doi.org/10.3390/a17090409.
25. Arif M, Fang G, Fida H, Musleh S, Yu D-J, Alam T. iMRSAPred: improved prediction of anti-MRSA peptides using physicochemical and pairwise contact-energy properties of amino acids. ACS Omega. 2024;9(2):2874–83. https://doi.org/10.1021/acsomega.3c08303.
26. Gribskov M, McLachlan AD, Eisenberg D. Profile analysis: detection of distantly related proteins. Proc Natl Acad Sci. 1987;84(13):4355–8. https://doi.org/10.1073/pnas.84.13.4355.
27. Li Y, et al. Robust and accurate prediction of self-interacting proteins from protein sequence information by exploiting weighted sparse representation based classifier. BMC Bioinformatics. 2022;23(S7):518. https://doi.org/10.1186/s12859-022-04880-y.
28. Yu B, et al. Prediction of subcellular location of apoptosis proteins by incorporating PsePSSM and DCCA coefficient based on LFDA dimensionality reduction. BMC Genomics. 2018;19(1):478. https://doi.org/10.1186/s12864-018-4849-9.
29. Waris M, Ahmad K, Kabir M, Hayat M. Identification of DNA binding proteins using evolutionary profiles position specific scoring matrix. Neurocomputing. 2016;199:154–62. https://doi.org/10.1016/j.neucom.2016.03.025.
30. Nanni L, Brahnam S, Lumini A. Wavelet images and Chou's pseudo amino acid composition for protein classification. Amino Acids. 2012;43(2):657–65. https://doi.org/10.1007/s00726-011-1114-9.
31. Wang X, Wang J, Fu C, Gao Y. Determination of corrosion type by wavelet-based fractal dimension from electrochemical noise. Int J Electrochem Sci. 2013;8(5):7211–22. https://doi.org/10.1016/S1452-3981(23)14840-1.
32. Lin K, Quan X, Jin C, Shi Z, Yang J. An interpretable double-scale attention model for enzyme protein class prediction based on transformer encoders and multi-scale convolutions. Front Genet. 2022;13:885627. https://doi.org/10.3389/fgene.2022.885627.
33 Lee H, Lee S, Lee I, Nam H. AMP-BERT : Prediction of antimicrobial peptide function based on a BERT model. Protein Sci. 2023;32(1):e4529. https://doi.org/10.1002/pro.4529.
34. Fryer D, Strümke I, Nguyen H. Shapley values for feature selection: the good, the bad, and the axioms. IEEE Access. 2021;9:144352–60. https://doi.org/10.1109/ACCESS.2021.3119110.
35. Heuillet A, Couthouis F, Diaz-Rodriguez N. Collective eXplainable AI: explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. IEEE Comput Intell Mag. 2022;17(1):59–71. https://doi.org/10.1109/MCI.2021.3129959.
36. Khan S, Khan M, Iqbal N, Dilshad N, Almufareh MF, Alsubaie N. Enhancing sumoylation site prediction: a deep neural network with discriminative features. Life. 2023;13(11):2153. https://doi.org/10.3390/life13112153.
37. Arif M, Musleh S, Fida H, Alam T. PLMACPred prediction of anticancer peptides based on protein language model and wavelet denoising transformation. Sci Rep. 2024;14(1):16992. https://doi.org/10.1038/s41598-024-67433-8.

38. Naeem M, Qiyas M, Abdullah L, Khan N, Khan S. Spherical fuzzy rough Hamacher aggregation operators and their application in decision making problem. AIMS Math. 2023;8(7):17112–41. https://doi.org/10.3934/math.2023874.
39. Qiyas M, Naeem M, Khan N, Khan S, Khan F. Confidence Levels bipolar complex fuzzy aggregation operators and their application in decision making problem. IEEE Access. 2024;12:6204–14. https://doi.org/10.1109/ACCESS.2023.3347043.
40. Zhu Z, Albadawy E, Saha A, Zhang J, Harowicz MR, Mazurowski MA. Deep learning for identifying radiogenomic associations in breast cancer. Comput Biol Med. 2019;109(March):85–90. https://doi.org/10.1016/j.compbiomed.2019.04.018.
41. Khan S, et al. Optimized feature learning for anti-inflammatory peptide prediction using parallel distributed computing. Appl Sci. 2023;13(12):7059. https://doi.org/10.3390/app13127059.
42. Khan S, Khan M, Iqbal N, Li M, Khan DM. Spark-based parallel deep neural network model for classification of large scale RNAs into piRNAs and Non-piRNAs. IEEE Access. 2020;8:136978–91. https://doi.org/10.1109/ACCESS.2020.3011508.
43. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90. https://doi.org/10.1145/3065386.
44. Khan S, AlQahtani SA, Noor S, Ahmad N. PSSM-Sumo: deep learning based intelligent model for prediction of sumoylation sites using discriminative features. BMC Bioinformatics. 2024;25(1):284. https://doi.org/10.1186/s12859-024-05917-0.
45. Khan S, Khan M, Iqbal N, Hussain T, Khan SA, Chou K-C. A Two-Level Computation model based on deep learning algorithm for identification of piRNA and their functions via Chou's 5-steps rule. Int J Pept Res Ther. 2020;26(2):795–809. https://doi.org/10.1007/s10989-019-09887-3.
46 Khan S, Khan M, Iqbal N, AmiruddinAbd Rahman M, Khalis Abdul Karim M. Deep-piRNA: Bi-layered prediction model for PIWI-interacting RNA using discriminative features. Comput Mater Contin. 2022;72(2):2243–58. https://doi.org/10.32604/cmc.2022.022901.
47. Khan S, et al. XGBoost-enhanced ensemble model using discriminative hybrid features for the prediction of sumoylation sites. BioData Min. 2025;18(1):12. https://doi.org/10.1186/s13040-024-00415-8.
48 Obadi A, AlHarbi A, Abdel-Razzak H, Al-Omran A. Biochar and compost as soil amendments: effect on sweet pepper (*Capsicum annuum L.*) growth under partial root zone drying irrigation. Arab J Geosci. 2020. https://doi.org/10.1007/s12517-020-05529-x.
49. Elsisi M, Mahmoud K, Lehtonen M, Darwish MMF. Reliable industry 4.0 based on machine learning and IoT for analyzing, monitoring, and securing smart meters. Sensors. 2021;21(2):487. https://doi.org/10.3390/s21020487.
50. Fawagreh K, Gaber MM, Elyan E. Random forests: from early developments to recent advancements. Syst Sci Control Eng. 2014;2(1):602–9. https://doi.org/10.1080/21642583.2014.956265.
51. Arif M, Fang G, Ghulam A, Musleh S, Alam T. DPI_CDF: druggable protein identifier using cascade deep forest. BMC Bioinform. 2024;25(1):145. https://doi.org/10.1186/s12859-024-05744-3.
52. Amjad A, Ahmed S, Kabir M, Arif M, Alam T. A novel deep learning identifier for promoters and their strength using heterogeneous features. Methods. 2024;230:119–28. https://doi.org/10.1016/j.ymeth.2024.08.005.
53. Ge F, et al. MMPatho: leveraging multilevel consensus and evolutionary information for enhanced missense mutation pathogenic prediction. J Chem Inf Model. 2023;63(22):7239–57. https://doi.org/10.1021/acs.jcim.3c00950.
54. Cheng D, Zhang S, Deng Z, Zhu Y, Zong M. kNN algorithm with data-driven k value. In: Luo X, Yu JX, Li Z, editors. Advanced data mining and applications. Cham: Springer International Publishing; 2014. p. 499–512.
55. Zhou G-P, Chen D, Liao S, Huang R-B. Recent Progresses in studying helix-helix interactions in proteins by incorporating the wenxiang diagram into the NMR spectroscopy. Curr Top Med Chem. 2015;16(6):581–90. https://doi.org/10.2174/1568026615666150819104617.


## Publisher's Note