BMC Medical Informatics
and Decision Making

RESEARCH Open Access

Optimizing protein sequence classification: integrating deep learning models with Bayesian optimization for enhanced biological analysis

Umesh Kumar Lilhore[1], Sarita Simiaya[1], Musaed Alhussein[2], Neetu Faujdar[3], Surjeet Dalal[4*] and Khursheed Aurangzeb[2]

Abstract
Efforts to enhance the accuracy of protein sequence classification are of utmost importance in driving forward biological analyses and facilitating significant medical advancements. This study presents a cutting-edge model called ProtICNN-BiLSTM, which combines attention-based Improved Convolutional Neural Networks (ICNN) and Bidirectional Long Short-Term Memory (BiLSTM) units seamlessly. Our main goal is to improve the accuracy of protein sequence classification by carefully optimizing performance through Bayesian Optimisation. ProtICNN-BiLSTM combines the power of CNN and BiLSTM architectures to effectively capture local and global protein sequence dependencies. In the proposed model, the ICNN component uses convolutional operations to identify local patterns. Captures long-range associations by analyzing sequence data forward and backwards. In advanced biological studies, Bayesian Optimisation optimizes model hyperparameters for efficiency and robustness. The model was extensively confirmed with PDB-14,189 and other protein data. We found that ProtICNN-BiLSTM outperforms traditional categorization models. Bayesian Optimization's fine-tuning and seamless integration of local and global sequence information make it effective. The precision of ProtICNN-BiLSTM improves comparative protein sequence categorization. The study improves computational bioinformatics for complex biological analysis. Good results from the ProtICNN-BiLSTM model improve protein sequence categorization. This powerful tool could improve medical and biological research. The breakthrough protein sequence classification model is ProtICNN-BiLSTM. Bayesian optimization, ICNN, and BiLSTM analyze biological data accurately.

Keywords Deep learning, Bayesian optimization, CNN, Bi-LSTM, Protein sequence classification, Bioinformatics

*Correspondence:
Surjeet Dalal
sdalal@ggn.amity.edu
[1]School of Computing Science and Engineering, Galgotias University, Greater Noida, UP, India
[2]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia
[3]Department of Computer Engineering and Applications, GLA University, 281406, UPMathura, India
[4]Amity University, Gurugram Haryana, India

Introduction

The distinctive spherical form of proteins originates from the unusual three-dimensional structure of polypeptides. This structure imparts features of proteins. The building blocks of proteins are amino acids. Amino acid residues are joined to create peptide bonds, which are the fundamental units of polymer chains [1]. By analyzing the DNA sequence, one can learn a protein's exact order of amino acids. Gene expression follows a predetermined sequence that is encoded in DNA. A gene is a separate DNA sequence. These locations hold the blueprints for certain genetic components, such as chromosomes, RNA molecules, or proteins [2]. Discovering and classifying protein structures and functions is tremendously challenging and complicated in bioinformatics. Conventional laboratory processes can process large amounts of RNA data. Proteins are classified into families and subfamilies meticulously, which helps researchers understand their functions in living creatures [3]. Fixing the problem will be easier if you do this.

Feature extraction approaches in traditional machine learning methodologies are responsible for protein molecule classification. However, how well manually produced features work is heavily dependent on the selection technique. Protein function prediction is achieved utilizing artificial neural networks (ANNs) parts of deep neural networks (DNNs). Discrete neural networks (DNNs) gradually enhance initial inputs as they go through the network's levels; these networks comprise multiple hidden layers. Conventional machine-learning approaches are laborious, time-consuming, and resource-intensive [4]. The main reason for these traits, described earlier, is the exponential growth of unique protein sequences.

Many different computer methods have been created because of the need to classify proteins and guess what biological functions they fulfill. Figure 1 shows the primary, secondary, tertiary, and quaternary groups of protein structures. It also shows several different protein structures. You can better understand the complicated world of proteins and the different groups they belong to with the help of this visual aid [5].

The careful selection of features is essential for classifying proteins in protein research. The outermost layer of the protein, the arrangement of its amino acid chains, and its functionalities all contribute to its unique qualities. Techniques seek to efficiently categorize structural protein molecules to facilitate a deeper understanding of evolutionary changes and their temporal relationships.

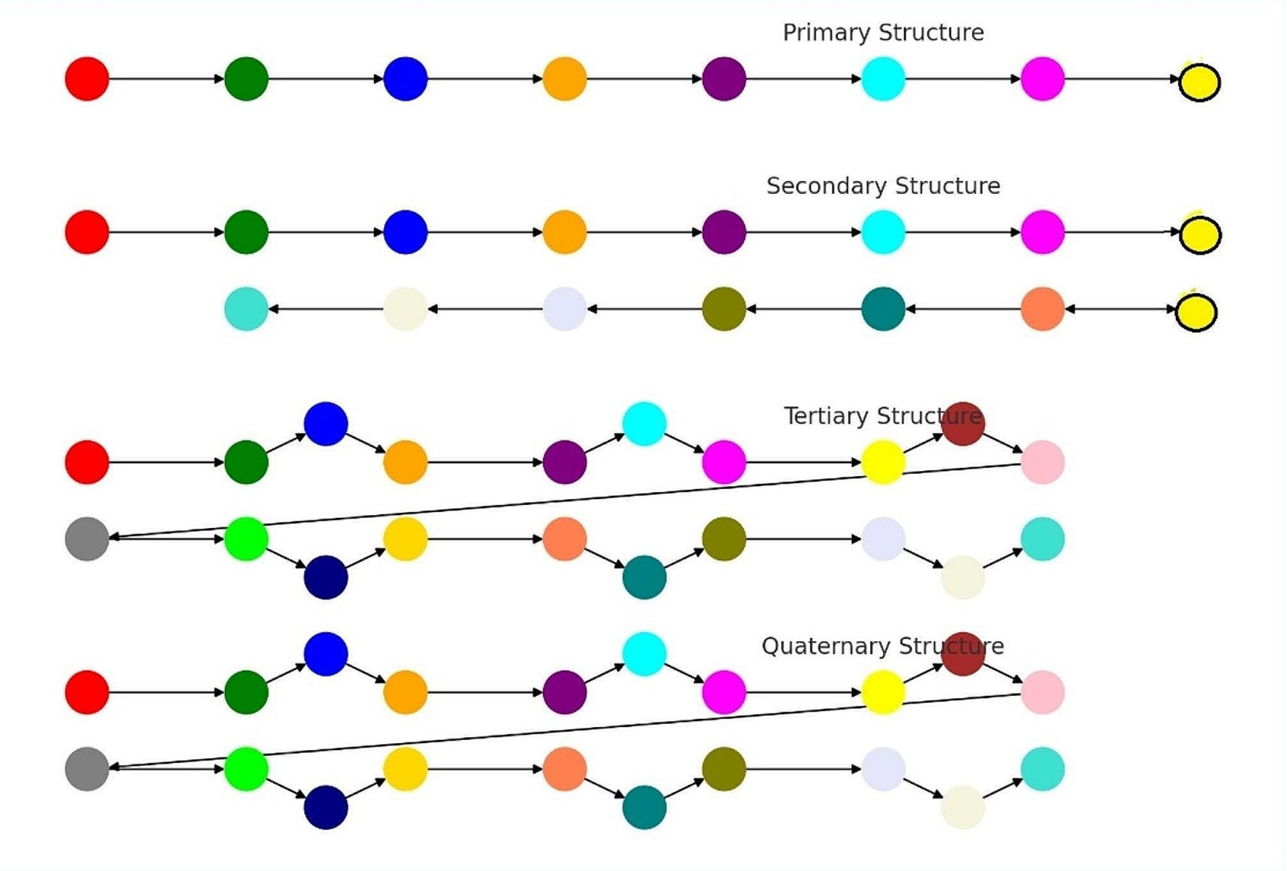


Fig. 1 Various protein structures (Primary, Secondary, Tertiary and Quaternary)

Advances in various techniques for identifying structural stability groups based on sequence details have been made possible by statistical analysis techniques [6].

DNN techniques have been widely embraced in modern scientific research, particularly in biomedical studies, to leverage recent advances in computational power. These techniques have been shown to outperform conventional bioinformatics methods in terms of effectiveness. Two areas in which they are commonly employed are the analysis of visual data and the application of machine learning to the processing of natural language. Single-task DNNs, which are responsible for making binary predictions, and multi-task DNNs, which can classify input data into multiple pre-defined classes, are the two categories used to classify DNNs [7]. Diverse classifications of DNNs are available to accomplish distinct functions in protein data modelling and analysis. The neural network architectures that have been discussed include convolutional neural networks, feed-forward neural networks, auto-encoder deep neural networks, deep belief networks, recurrent neural networks, restricted Boltzmann machines, and graph convolutional networks. Developing improved methods for accurately categorizing protein sequences in proteomics research is the main goal of this work [8].

This study tackles two key research questions: (1) overcoming difficulties in protein sequence classification approaches and (2) evaluating feature extraction methods with existing CNN, LSTM, BiLSTM, and ProtICNN-BiLSTM models [9]. Both of these questions are important in the field of research. Both of these concerns may need to be addressed in subsequent research. When advancing medical research and biological analysis, raising the bar on the precision of protein sequences is necessary. Within the scope of this investigation, an improved model known as ProtICNN-BiLSTM is presented. Attention-Based Improved Convolutional Neural Networks are paired with Bidirectional Long Short-Term Memory units in this particular instance. The model utilizes the Bayesian Optimisation technique to capture the local and global interactions within protein sequences. This is done to improve the accuracy of the protein sequence categorization process. The remarkable performance of the model in comparative investigations reveals its major influence on medical and biological research [10]. In addition to its promise for enhancing protein sequence categorization, the model also demonstrates its promise for improving classification.

A breakdown of the article's structure is as follows: the Related Work section is where we take a detailed look at the previous research that has been done. The Materials and Methods section presents a comprehensive description of the methods and materials used for the investigation. In the Experimental Results and Discussion section, a summary of the findings and an evaluation of the significance of those findings are offered. For the time being, the study has been completed, and Conclusion and Future Directions section provides an outline of potential future research areas.

Related work

The newest protein structure analysis and classification deep learning algorithms are reviewed here. It includes hybrid models and attention mechanisms. Paper [1] used several indicators to compare deep-learning protein sequence synthesis algorithms. This research examined many cutting-edge deep-learning approaches for protein sequence generation. Many indicators were used to compare deep learning methods. We explored these strategies for synthesizing unique protein sequences from various sequences. The study's main contribution is comparing each approach's merits and cons. A diversified protein sequence library ensured complete research. Some approaches succeeded; however, there were few evaluation criteria and methodologies, so protein sequence design concerns may have been missed.

A deep learning-based method for predicting proteins of snake toxins using word embeddings is proposed in the article [2]. This study accomplishes the prediction of proteins containing snake venom through word embeddings within a deep learning framework. The model learned to link sequence patterns with toxin characteristics from a collection of annotated snake toxin proteins. With impressive performance indicators such as a 99.71% average 6-class classification success rate and a 99.85% binary classification accuracy for SARS-CoV-2 compared to HIV-1, Deep-STP proved to be highly accurate in predicting these proteins. Notwithstanding these encouraging findings, the model's performance may be affected by the variety and quality of the training dataset as well as particular architectural decisions.

In [3], the authors review and investigate the many methods and instruments available for determining the locations of lysine malonylation sites in protein sequences. Machine learning and deep learning are essential to these methods and resources. We tested machine learning and deep learning algorithms to predict lysine malonylation. Researchers tested how well these algorithms predicted published protein sequence positions. This book's most essential section evaluates the latest abilities and instruments, highlighting their benefits and cons. Every protein sequence in the collection has its lysine malonylation sites determined. Despite finding some promising solutions, the research could not evaluate their generalizability because of annotated data quality and accessibility issues.

The MaTPIP technique was initially introduced by the authors of the work described earlier [4]. Utilizing a deep learning architecture, it provides eXplainable artificial intelligence to predict sequence-driven, feature-mixed protein-protein interactions. As part of its deep learning architecture, MaTPIP makes use of explainable artificial intelligence approaches. This work aimed to attempt to predict the interactions between proteins.

The algorithm learned protein sequences and interactions. Sequence characteristics and other inputs helped it predict interactions. An explanation-based AI system is MaTPIP's main contribution. This knowledge helps us predict protein interactions. Positive findings showed good interaction prediction accuracy during the investigation. Protein interaction complexity and annotated data may affect model efficiency. Deep learning makes protein syntheses interactive [5]. They proved that their model successfully predicted health system protein-protein interaction locations. They also acknowledged that machine learning was the primary component that led to this accomplishment. In this article, we offer a model that uses machine learning to forecast the times at which protein-protein interactions occur. Training on a set of protein sequences and their interaction sites allowed the program to predict these sites using sequence information. Including practical contact site prediction is a big improvement. This approach may benefit healthcare systems. The experiment showed exact protein-protein interaction sites. This was found throughout the investigation. However, the study found several factors may alter the model's efficacy.

These variables include the degree of interaction complexity and the availability of data that has been annotated. Six deadly RNA viruses are being studied [6]. Their feature-engineered protein patterns classify these viruses into several types. The viruses are "Human respiratory virus type 3, influenza A, B, and C, and HIV-1". Another respiratory virus is HIV-1. Data classification and analysis using linear complexity measures. Because of this, it provided reliable categorization, especially for large data sets. The impressive average success rate of 99.71% that the model achieved in identifying the six classes included in the data set is comparable to a published method and implemented with a high degree of precision. The SARS-CoV-2 binary classification fared better than the HIV-1 classification, with a success rate of 99.85%. This was the case across all of the success rates. Additionally, a convolutional neural network (CNN) and a gated recurrent unit (GRU) can be utilized in conjunction with a long short-term memory (LSTM) to locate proteins that bind to DNA.

This is one of several innovative ways developed in recent years. CNN-BiLG describes this combination. CNN-BiLG collects more data and analyzes protein sequence contextual relationships more thoroughly than previous approaches.

An explanation for this can be found in the enhanced capability of the CNN-BiLG technique to capture detailed information. Compared to deep learning and machine learning predictions, CNN-BiLG demonstrated superior performance, as demonstrated by the results of the studies. According to the validation findings, the reports have an incredible accuracy of 95%.

According to many investigations, the suggested model outperforms previous methods in efficiency, cost-effectiveness, and classification accuracy. The proposed model is cheaper.

NLP-based text categorization methods are widely used to classify protein sequences [8]. Deep learning and word embedding have improved text categorization. These advancements have increased protein categorization accuracy and opened new choices. Word-embedded protein sequence representations encounter many challenges in natural language processing (NLP) because amino acid sequences have different "words" than other sequences.

The longer sequences and smaller letter sizes included in the protein data bring additional difficulties for the learning models. This can be attributed to the presence of lengthier sequences in protein data. It has been established that pre-training is one way that can help boost the effectiveness of machine learning techniques. Even though it was initially proposed [9] for computer vision applications, it is currently being utilized extensively in a wide range of machine learning applications, including those connected to language.

Research shows that pre-trained models offer high generalization and convergence rates for tasks with limited training data. Pre-training methods like BERT and ELMo are important despite processing resource constraints. Data-driven neural networks like GCNs can delay hidden cell interactions. To learn and remember, GCNs act sequentially in biology. Hidden cell connections in data-driven Graph Convolution Networks (GCNs) are time-delayed. GCNs build memories and integrate knowledge by operating in biological sequences one element at a time.

Techniques such as SeqVec and ProtTrans [10] use language models and transformer frameworks to represent protein sequences as embedding vectors. This helps to contribute to the understanding of the biophysical characteristics of proteins. Pre-training can leverage comprehensive labelled datasets and transfer knowledge to smaller data problems due to shared pattern characteristics in protein sequence-based classification tasks. This is in contrast to deep learning models derived from natural language processing contexts, which require significant computing power.

Many different types of biological data [11] can be used to predict the functions of proteins. These data types include sequences, three-dimensional structures, folding information, protein-protein interactions, variations in gene expression, amino acid families, and their integration. The classification of common data has been accomplished through the development of statistical theories through the use of techniques such as decision trees, Support Vector Machines (SVM) [12], and Neural Networks (NN) [13]. A few studies that applied SVM after feature extraction from protein sequences demonstrated the potential of SVM in protein classification. Deep learning methods have shown promise in investigations focusing on relatively small groups of proteins and functional categories. Although these methods have not been extensively explored for large-scale protein function prediction pipelines [14], DNN architectures have been trained to anticipate protein operations through research investigations conducted using various protein characteristics. These investigations included both single-tasking and multi-tasking architectures. Table 1 presents a comparison of various existing research in protein sequence analysis.

Materials and methods

This section initiates by introducing the datasets employed in the model's development. Subsequently, this article elucidates the conceptual framework and testing methodologies. Finally, a model algorithm utilized in the demonstration is presented.

Proposed hybrid model

ProtICNN-BiLSTM is a proposed hybrid model that combines Bidirectional Long Short-Term Memory (BiLSTM) units with Improved Convolutional Neural Networks (ICNN) and uses Bayesian Optimization [23, 24] to improve the model parameters. Using the strengths of the ICNN and BiLSTM frameworks, the ProtICNN-BiLSTM method efficiently captures both local and global interdependencies in protein sequences. Bayesian optimization for hyperparameter modifications leads to an even greater increase in model performance. The operation of every component is explained in depth in this part, together with the pertinent equations [25]. The suggested hybrid model ProtICNN-BiLSTM is architecturally illustrated in Fig. 2. Following its receipt by the input layer, the protein sequence data is sent through a convolutional layer with 64 filters, a 3×3 kernel size, a stride of 1, and "same" padding. ReLU activation comes last after batch normalization. Another convolutional layer with 128 filters, a 3×3 kernel size, a stride of 1, and "same" padding processes the output of this layer. Batch normalization follows [26].

ICNN model

The ICNN part of the ProtICNN-BiLSTM method is supposed to extract local properties from the protein sequences. Convolutional Neural Networks can efficiently capture spatial hierarchies in data using convolutional processes [27]. ICNN applies attention methods to enhance feature extraction. Many improvements are included in the enhanced CNN architecture of the ProtICNN-BiLSTM model to increase efficiency and capture more intricate features from protein sequences. These improvements have included the remaining connections to address the disappearance of gradients and enhance gradient flow during training. Batch normalization layers are added after each convolutional layer to standardize the input of individual layers, therefore stabilizing and speeding up the training process even further [28].

Table 1 Comparison of various existing research in protein sequence analysis

References	Method used	Dataset	Outcome	Future scope
[15]	Attention-based Neural Network	PDB-14,189	Improved classification accuracy	Investigate attention mechanisms in larger datasets
[16]	RNNs	DNA Data Bank of Japan	Improved Classification	Exploration of hybrid models in proteomics research
[17]	Hybrid CNN-RNN Model	European Nucleotide Archive	Enhanced protein structure prediction	Integration of attention mechanisms in drug design
[18]	Transformer Networks	The Consensus CDS protein set database	Better classification and performance	Exploration of hybrid models in proteomics research
[19]	Attention-based Hybrid Model	PDB-2272	State-of-the-art performance	Application of hybrid models in drug target prediction
[20]	Random Forest and Decision Tree	SWISS-PROT Dataset	Captured spatial dependencies	Investigation of attention mechanisms in protein engineering
[21]	CNN and RNN	PROSITE database	Improved classification accuracy	Application of hybrid models in drug target prediction
[22]	NLP with Machine learning	CASP Dataset	Improved classification accuracy	Development of attention mechanisms for protein-protein interaction prediction
Proposed Hybrid	Attention method with Improved CNN and BiLSTM	PDB-14,189 Datasets	Improved classification accuracy	Time Complexity can improve.

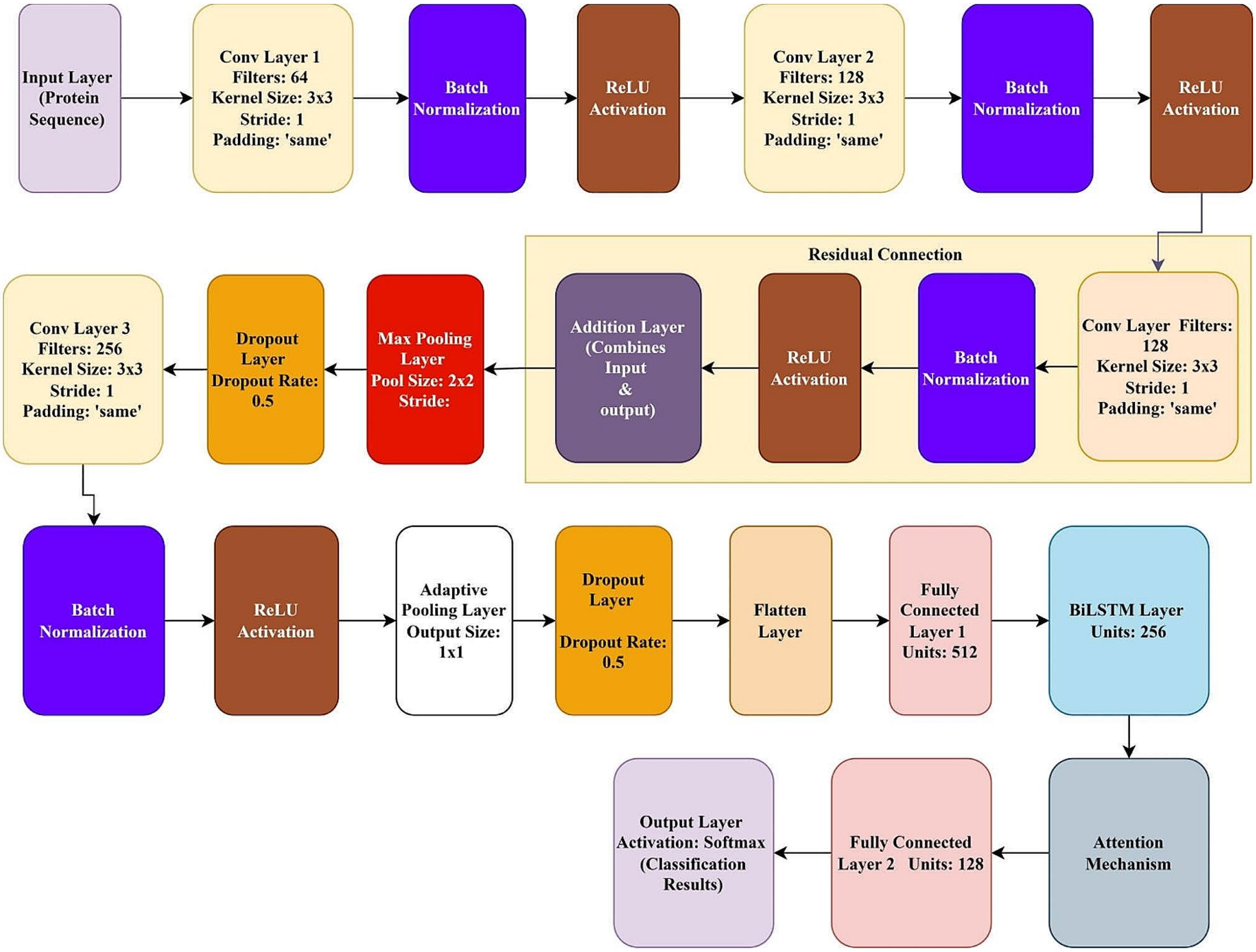

Input Layer (Protein Sequence)
Conv Layer 1 Filters: 64 Kernel Size: 3x3 Stride: 1 Padding: 'same'
Batch Normalization
ReLU Activation
Conv Layer 2 Filters: 128 Kernel Size: 3x3 Stride: 1 Padding: 'same'
Batch Normalization
ReLU Activation
Residual Connection
Conv Layer Filters: 128 Kernel Size: 3x3 Stride: 1 Padding: 'same'
Batch Normalization
ReLU Activation
Addition Layer (Combines Input & output)
Max Pooling Layer Pool Size: 2x2 Stride:
Dropout Layer Dropout Rate: 0.5
Conv Layer 3 Filters: 256 Kernel Size: 3x3 Stride: 1 Padding: 'same'
Batch Normalization
ReLU Activation
Adaptive Pooling Layer Output Size: 1x1
Dropout Layer Dropout Rate: 0.5
Flatten Layer
Fully Connected Layer 1 Units: 512
BiLSTM Layer Units: 256
Attention Mechanism
Fully Connected Layer 2 Units: 128
Output Layer Activation: Softmax (Classification Results)

Fig. 2 Architecture of proposed hybrid model (ProtICNN-BiLSTM)

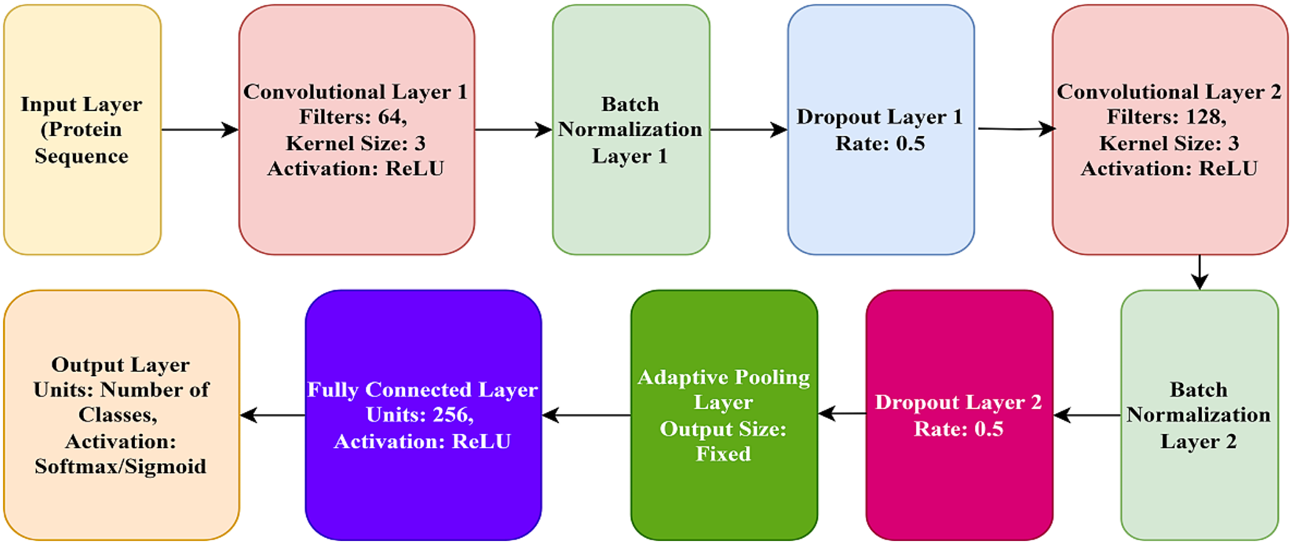

Input Layer (Protein Sequence
Convolutional Layer 1 Filters: 64, Kernel Size: 3 Activation: ReLU
Batch Normalization Layer 1
Dropout Layer 1 Rate: 0.5
Convolutional Layer 2 Filters: 128, Kernel Size: 3 Activation: ReLU
Batch Normalization Layer 2
Dropout Layer 2 Rate: 0.5
Adaptive Pooling Layer Output Size: Fixed
Fully Connected Layer Units: 256, Activation: ReLU
Output Layer Units: Number of Classes, Activation: Softmax/Sigmoid

Fig. 3 Architecture of improved CNN model

ReLU activations are included following batch normalization to increase the expressiveness of the model even further and introduce non-linearity. Half-rate dropout layers train by arbitrarily removing certain input units to prevent further overfitting. Consistently handling sequences of varying lengths is easier using adaptive pooling layers, ensuring a constant output size irrespective of input dimensions. These enhancements collectively increase the overall efficacy of the ProtICNN-BiLSTM model by enhancing CNN's comprehension of flexible and robust features from protein sequences [29].

The Improved CNN Model's design is shown in Fig. 3. The revised CNN architecture receives protein sequences as numerical array input to do protein sequence analysis. The layered layout of the proposed paradigm is depicted in Fig. 4. 64 The first convolutional layer uses 3×3 filters to search for local patterns in the sequences. Next, the ReLU activation function is used to introduce non-linearity. A dropout layer with a rate of 0.5 is employed to reduce overfitting, and batch normalization is used to stabilize the activations [30].

When it comes to the second convolutional layer, there are 128 filters, each of which measures three by three measures. This particular layer is responsible for capturing more abstract and complex qualities. The ReLU activation function is utilized to achieve non-linearity. Next is a dropout layer, followed by batch normalizing. These phases promote long-term, versatile learning. Since adaptive pooling resizes feature maps to a preset size, it allows for many input periods. The 256-unit fully connected layer can combine these characteristics to build sophisticated and abstract representations using the Rectified Linear Unit (ReLU) activation function [31].

In order to compute the probability of the protein classes, the output layer of the multi-class classification process uses the application of the SoftMax activation function. This is done to achieve the desired results. The classification procedure is carried out to guarantee that it is appropriate. In addition, local and global feature extraction procedures have been incorporated into this architectural design [32] to improve the precision and continuity of the protein sequence categorization process. Along with regularization strategies, this design also includes regularization techniques.

Convolution operation process The convolution process is used to extract local features from the sequence that is being input [33]. The mathematical form of the convolution operation is summarized in Eq. (1), which is presented below. $Z_{i,j}^{k}$ represents the output of the convolution operation phase, (i, j) represents positions, k represents filter, b^{k} is a bias term, M and N filter size.

$$Z_{i,j}^{k} = \sum {}_{m=1}^{M} \sum {}_{n=1}^{N} x_{i+(m-1).\ (j+n-1) W_{m,n}^{k} + b^{k}} \tag{1}$$

Attention mechanism An attention technique has been implemented to concentrate on the most important features retrieved by the convolutional layers [34]. According to Eq. (2), the attention weights are calculated as a given. Here σ_i: attention weight, (e_i): Energy score, L: Local feature count.

$$\sigma_i = \frac{\exp\left(e_i\right)}{\sum {}_{\mathrm{j}=1}^{\mathrm{L}} \exp\left(e_j\right)} \tag{2}$$

Energy score calculation *An energy score*e_i *can be calculated by* Eq. (3). *Here*W_a *and*b_a *are learning parameters*, h_i : *hidden state* [35].

$$e_i = tanh(W_a * h_i + b_a) \tag{3}$$

Weighted feature representation The attention weights and the convolutional features are combined to produce the weighted feature representation W_f as described in Eq. (4). HereW_f: weight feature, h_i : *hidden state, and* L: Local feature count.

$$W_f = \sum {}_{i=1}^{L} \sigma_i h_i \tag{4}$$

Bidirectional long short-term memory (BiLSTM)

Bidirectional Long Short-Term Memory, also known as BiLSTM, is a more advanced kind of Recurrent Neural Network (RNN) that was developed to handle sequential data, for instance, sequences of protein, by taking into account dependencies in both the forward and backward directions [36]. The operation of BiLSTM is as follows.

Forward-LSTM (Fw-LSTM) Predominantly, the Forward LSTM Layer The input sequence is processed from the beginning to the end, with the forward dependencies being captured. Each cell in this layer comprises three gates: input, forget, and output [37]. These gates control the flow of information, enabling the network to remember or forget knowledge from the past depending on the circumstances; the essential formulas are presented from Eq. (5) to (10). Table 2 presents the key symbols used in forward-BLSTM.

$$f_t^{(f)} = \ \{W_t^{(f)} \left[h_{t-1}^{(f)},\ x_t\right] +\ b_f^{(f)}\} \tag{5}$$

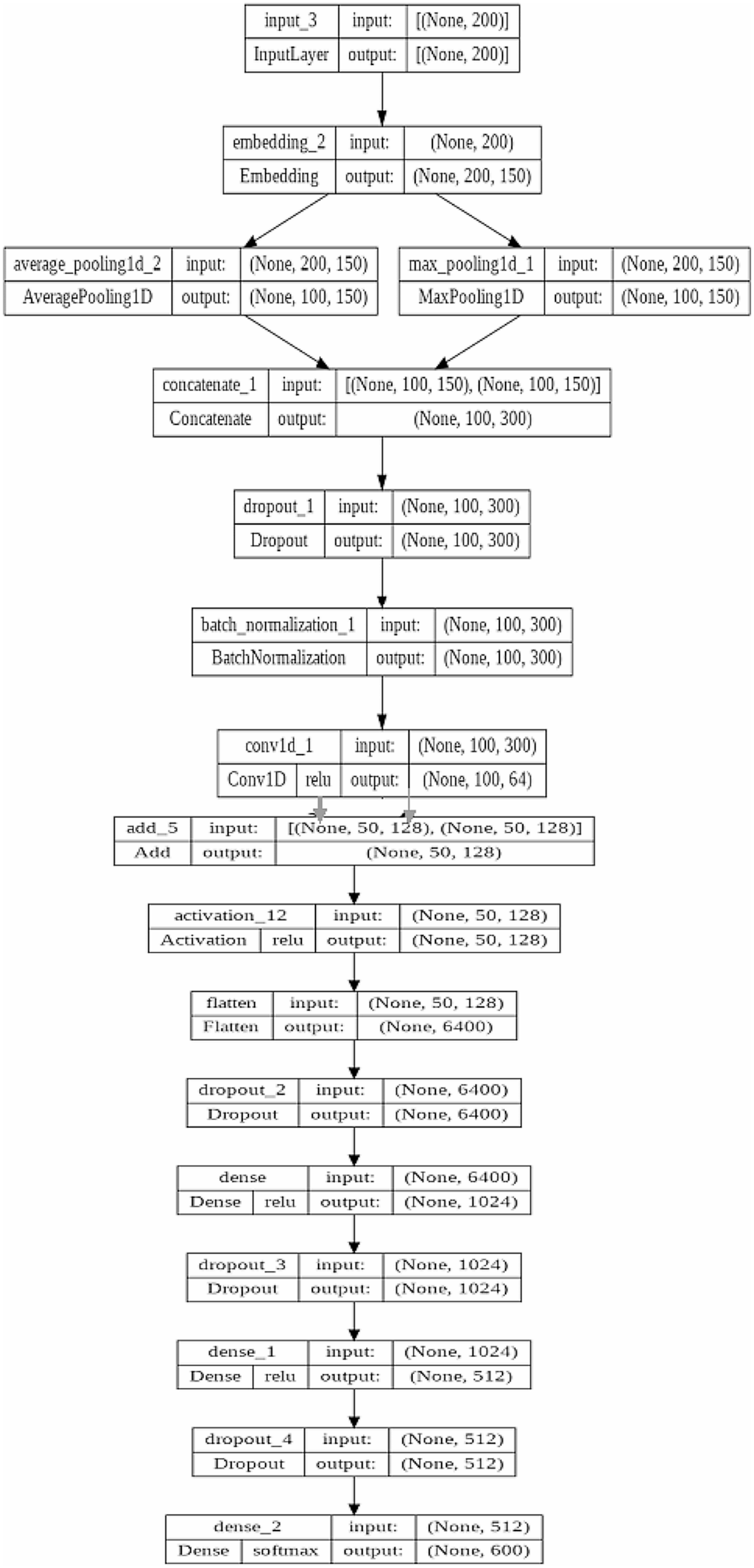

input_3 | input: | [(None, 200)]
InputLayer | output: | [(None, 200)]
embedding_2 | input: | (None, 200)
Embedding | output: | (None, 200, 150)
average_pooling1d_2 | input: | (None, 200, 150)
AveragePooling1D | output: | (None, 100, 150)
max_pooling1d_1 | input: | (None, 200, 150)
MaxPooling1D | output: | (None, 100, 150)
concatenate_1 | input: | [(None, 100, 150), (None, 100, 150)]
Concatenate | output: | (None, 100, 300)
dropout_1 | input: | (None, 100, 300)
Dropout | output: | (None, 100, 300)
batch_normalization_1 | input: | (None, 100, 300)
BatchNormalization | output: | (None, 100, 300)
conv1d_1 | input: | (None, 100, 300)
Conv1D | relu | output: | (None, 100, 64)
add_5 | input: | [(None, 50, 128), (None, 50, 128)]
Add | output: | (None, 50, 128)
activation_12 | input: | (None, 50, 128)
Activation | relu | output: | (None, 50, 128)
flatten | input: | (None, 50, 128)
Flatten | output: | (None, 6400)
dropout_2 | input: | (None, 6400)
Dropout | output: | (None, 6400)
dense | input: | (None, 6400)
Dense | relu | output: | (None, 1024)
dropout_3 | input: | (None, 1024)
Dropout | output: | (None, 1024)
dense_1 | input: | (None, 1024)
Dense | relu | output: | (None, 512)
dropout_4 | input: | (None, 512)
Dropout | output: | (None, 512)
dense_2 | input: | (None, 512)
Dense | softmax | output: | (None, 600)

Fig. 4 Layers and parameters of the proposed hybrid model

Table 2 Key symbols used in Forward BLSTM

Symbol	Description
$\boldsymbol{f_t^{(f)}}$	*Forget the gate at the time step for the forward LSTM.*
σ	*Sigmoid activation function*
$\boldsymbol{W_f^{(f)}}$	*Weight matrix for the forget gate*
$\boldsymbol{h_{t-1}^{(f)}}$	*Hidden state at the previous time step t – 1for the forward LSTM*
$\boldsymbol{x_t}$	*Input at time step t*
$\boldsymbol{b_f^{(f)}}$	*Bias term for the forget gate*
$\boldsymbol{Input_t^{(f)}}$	*Input gate at time step t for the forward LSTM*
$\boldsymbol{W_{input}^{(f)}}$	*Weight matrix for the input gate*
$\boldsymbol{b_{input}^{(f)}}$	*Bias term for the input gate*
$\boldsymbol{b_{ouput}^{(f)}}$	*Output gate at time step t for the forward LSTM*
$\boldsymbol{W_{output}^{(f)}}$	*Weight matrix for the output gate*
$\boldsymbol{b_{output}^{(f)}}$	*Bias term for the output gate*
$\boldsymbol{\check{C}_t^{(f)}}$	*Candidate cell state at time step t for the forward LSTM*
tanh	*Hyperbolic tangent activation function*
$\boldsymbol{W_C^{(f)}}$	*Weight matrix for the candidate cell state*
$\boldsymbol{b_C^{(f)}}$	*Bias term for the candidate cell state*
σ	*Cell state at time step t for the forward LSTM*
$\boldsymbol{C_{t-1}^{(f)}}$	*Cell state at the previous time step t – 1t-1t – 1 for the forward LSTM*
$\boldsymbol{h_t^{(f)}}$	*Hidden state at time step t for the forward LSTM*

Table 3 Instance count in PDB-14,189

Dataset	Total Instances	Positive instance	Negative Instance
PDB-14,189	14,189	7129	7060

$$Input_t^{(f)} = \{W_{input}^{(f)}\left[h_{t-1}^{(f)},\ x_t\right] + b_{input}^{(f)}\} \tag{6}$$

$$Output_t^{(f)} = \{W_{output}^{(f)}\left[h_{t-1}^{(f)},\ x_t\right] + b_{output}^{(f)}\} \tag{7}$$

$$\check{C}_t^{(f)} = tanh\,\{W_C^{(f)}\left[h_{t-1}^{(f)},\ x_t\right] + b_C^{(f)}\} \tag{8}$$

$$C_t^{(f)} = f\,_t^{(f)}\left[C_{t-1}^{(f)} + input_{t-1}^{(f)}\right]\ \check{C}_t^{(f)}\} \tag{9}$$

$$h_t^{(f)} = output\,_t^{(f)}[\tanh(\ \check{C}_t^{(f)}) \tag{10}$$

Backward-LSTM (Bw-LSTM) An additional LSTM layer simultaneously processes the sequence from the end to the beginning, capturing the backward dependencies. This layer functions in a manner that is analogous to that of the forward layer, employing the same gate mechanisms to control the flow of information [38].

Cumulative output Concatenating hidden states from forward and backward LSTM layers every time step. With information on each component and consideration for the past and future, this combination delivers a complete sequence picture every time. For the protein sequence "BHDU," the forward LSTM would be B◊H◊ D◊U. In contrast, the reverse LSTM works as follows: U◊D◊H◊B. Combining the concealed state from both sides ensures that every point in the sequence incorporates information from the previous and next portions [39]. BiLSTM is effective for complex sequential data like protein sequences, where component interactions determine analysis and classification. Its bidirectional approach causes this.

Bayesian optimization method

Bayesian optimization allows one to modify the hyperparameters in complex processes such as deep learning models. Optimized hyperparameter setups are found by constantly updating a probabilistic model and balancing exploration and exploitation. Successfully and carefully modifying hyperparameters, Bayesian optimization improves ProtICNN-BiLSTM protein categorization [40]. Bayesian optimization operates in the model as follows.

- **Defining an Objective Function**: The goal function f(x) is our attempt to maximize the performance metric for protein sequence analysis. Equation 11 is another classification metric; precision, recall, or accuracy are additional possibilities.

$$f\,(x) = Accuracy(ProICNN\_BiLSTM(x\,)) \tag{11}$$

- **Set a Gaussian Process**: To get a close approximation to the objective function, a Gaussian Process (GP) is started. A mean function μ(x) and a covariance function Cf(x, x′) are utilized by the GP model to forecast the value and uncertainty of the objective function (Eqs. 12 and 13).

$$\varPhi\ (x) = Mean\_Function \tag{12}$$

$$\mathrm{Cf(x, x\prime\,)}\ = Covariance\_Function \tag{13}$$

- **Define an Acquisition Function**: The selection of the subsequent hyperparameter point to examine is accomplished with the assistance of the acquisition function. With its ability to balance exploration and exploitation, the Expected Improvement (EI) function is a popular option (Eq. 14).Here$\beta\ (x)$: Acquisition function, x^{+}: Selected best hypothetical parameter at the current point.

$$\beta\ (x) =\ E[\max(0, f\ (x) - f\ \left(x^{+}\right)\ )] \tag{14}$$

- **Hyperparameter Update**: Replace the old GP model with the updated one using the evaluation data. This makes the GP model's goal function approximation more accurate.

Algorithm proposed hybrid model

The key steps for the proposed hybrid model are described in Algorithm 1 below.

```
Algorithm 1: Proposed Hybrid model for protein sequence
Input: Protein dataset
Output: Protein Sample categories with different classes.
Step 1: Import and preprocess the data
1. Import data on protein sequences through the Protein Data Bank samples and another pertinent resource.
2. Perform preprocessing on the patterns, considering features such as amino acid structure, physicochemical characteristics, and structural details.
Step 2: Divide the data
1. Partition the sample among training and testing collections with an 80:20 ratio, guaranteeing that the protein groups are evenly distributed in both sets.
Step 3: Architectural Design
1. Create a Hybrid Convolution Neural Network (CNN)-Bidirectional Long Short-Term Memory (BiLSTM) model to capture short, practical- and longer-range relationships in protein patterns.
2. Convolution layers can be used to obtain spatial patterns, whereas Bi-LSTM layers, in combination, should be used to retrieve sequential data.
Step 4: Hyperparameter Tuning
1. Specify a range of hyperparameter values, including learning rates, batch size, convolutional filtering size, LSTM units, and dropout rates.
2. Implement Bayesian Optimisation to systematically and effectively investigate and identify the algorithm's most optimal hyperparameter settings.
Step 5: Assessment Criterion:
1. Select the F1-score as the primary evaluation metric for precision and recall, essential in imbalanced protein sequence datasets.
Step 6: Training and evaluating the model:
1. Utilise Bayesian optimization to obtain the optimal Hyperparameters and then train the model on the training set.
2. Assess the model's performance on the test set, considering metrics such as accuracy, precision, recall, F1-score, and any metrics specific to the domain.
Step 7: Analysis and depiction:
1. Observe the model's performance and examine instances where it incorrectly classified data.
2. Utilise interpretability tools to gain insights into the specific portions of the sequences that have the most significant impact on predictions.
```

Dataset

This research utilizes the standard protein online dataset PDB-14189 (Protein Data Bank-14189) [31]. The dataset comprises a heterogeneous collection of patterns of protein derived from different organisms, which includes "enzymes, antibodies, structural proteins, transport proteins, receptors, and other functional categories. The description of each protein sequence includes details on its components, operation, and biological characteristics. Such information may involve the secondary structure components, binding of legend sites, protein class categorization, organism site, the procedure utilized for organization commitment, and clarity of the empirical form. The PDB dataset is frequently utilized in bioinformatics and molecular science studies for various objectives, such as protein structural estimation, multifunctional annotation, interaction between protein and protein prediction, chemical effects target recognition, and automated tasks such as classification.

Among the 14,189 cases in the PDB-14189 dataset, there are 7,129 positive instances of DNA-binding proteins and 7,060 negative instances of proteins that do not bind DNA. When researching bioinformatics and machine learning, this dataset is frequently utilized for protein function prediction and structural analysis tasks. Table 3 presents the PDB dataset description.

Data pre-processing

Experiments and protein databases were two reliable sources from which the dataset was initially carefully assembled. Thus, the process began. We next enthusiastically launched a thorough data cleaning process. This included removing duplicate sequences, correcting errors, and meticulously handling missing data. By now, the integrity and dependability of the dataset had to be ensured for the following analysis [41].

The feature extraction was achieved using advanced techniques to convert the protein sequences into numerical representations. One-hot encoding was used to turn every amino acid in the sequence into a binary vector, achieving increased Specificity. We have recovered a large range of amino acid physicochemical properties to enhance the feature set and capture significant features of the proteins. Among these were a specific molecular mass and water repellency. Oversampling techniques were used to alleviate class imbalance. Class imbalance resulted when there were more negative than positive instances, that is, proteins that did not link to DNA than positive examples did. It needed to make fictitious data points for the minority class to achieve a balanced distribution and remove bias from the machine learning algorithms.

The dataset was next separated into training, validation, and test sets. Class distribution was carefully maintained inside each subset to ensure accuracy. The models were trained using the training set throughout the machine learning process. The validation set helped us select the best model and adjust the hyperparameters. Finally, the test set assessed the whole model's performance. We last used feature scaling techniques like normalization to ensure that all features were equivalent. The best efficiency and ability of the machine learning algorithms to learn from the dataset were guaranteed by this stage. These preprocessing approaches ensured that the protein data was prepared and optimally suited for the training and analysis of machine learning models. Results were, therefore, more precise and trustworthy.

Performance measuring parameters

Key parameters evaluated the suggested and present models' prediction accuracy. Precision (P), recall (R), F1-score, support, specificity (SPC), sensitivity (SNS), Matthews' correlation coefficient (MCOC), and accuracy (ACR) are calculated using Eq. 13 to 16 [42]. Here, TP: True positive, FN: False Negative, TN: True Negative and FP: False positive.

Precision

Precision is calculated as the count of true positives divided by the overall quantity of positive cases discovered, as stated in Eq. 15.

$$P = \frac{TP}{[TP + FP]} \tag{15}$$

Recall

Divide the number of successfully identified positive observations by the total positive specimens to compute recall as presented by Eq. 16.

$$\boldsymbol{R} = \frac{\boldsymbol{TP}}{[\boldsymbol{TP} + \boldsymbol{FN}]} \tag{16}$$

Table 4 Hyperparameters used in experiments

Hyperparameters	Value	Details
Batch Size	128	Data size used for training in the model
Epoch	1500	The complete pass used for model training
Optimizer	ADAM optimizer	The algorithm which updated the weights of the model during the training phase
Rate of Learning	0.001	It mainly defines the step size, which model eight will update during training.
Total Convolution Layers	3	Building Block of CNN
Total Number of Fully Connected Layers	2	It performs the classification task

F1-score

In binary classification (and multi-class categorization), the F1-score measures precision and recall (Eq. 17).

$$\boldsymbol{FS} = 2 \times \left\{ \frac{[\boldsymbol{P} \times \boldsymbol{R}]}{[\boldsymbol{P} + \boldsymbol{R}]} \right\} \tag{17}$$

Specificity

As shown in Eq. 18, SPC is a binary categorization metric that measures a model's negative case detection accuracy.

$$\boldsymbol{FS} = 2 \times \left\{ \frac{[\boldsymbol{P} \times \boldsymbol{R}]}{[\boldsymbol{P} + \boldsymbol{R}]} \right\} \tag{18}$$

Accuracy

Eq. 18 calculates accuracy by dividing the number of successfully predicted instances by the total number of occurrences.

$$\mathrm{ACR} = \frac{[[\mathrm{TP} + \mathrm{TN}}{[\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}]} \tag{19}$$

Experimental results and discussion

The Proposed and existing models are implemented using Python programming, and various performance-measuring parameters are calculated. Different performance metrics were computed to evaluate the efficacy of these models. The study leveraged PyTorch, an openly accessible deep-learning library. The proposed hybrid model was developed using Python Keras [43]. Evaluation of absolute parameters involved the use of 'PDB-14189'. The dataset was divided into an 80% training sample and a 20% testing sample [44–47].

Hyperparameter specification

Table 4 represents the hyperparameter details. Table 4 defines the Hyperparameters used in experimental analysis [48–50]. Table 5 contains the parameters elements of CNN and LSTM in the proposed hybrid model, which affect and result from the shape of every level of a DNN in protein analysis.

Results for different parameters

The PDB-14,189 dataset was used as a key dataset in this research. Various performance measuring parameters were calculated to measure the performance of existing CNN, CNN-LSTM and the proposed hybrid model. Figure 5 presents protein sequence patterns for different classes. Figure 5 presents (a) the protein sequence frequency of attributes, 5(b) presents the protein sequence

Table 5 Parameters details of CNN and BiLSTM in the proposed hybrid model

Layer (Type)	Parameters	Output Shape
embedding (Embedding)	650	(None, 350, 25)
conv1d_1 (Conv1D)	12,928	(None, 350, 128)
conv1d_2 (Conv1D)	82,048	(None, 346, 128)
average_pooling1d_1	0	(Average (None, 173, 128)
conv1d_3 (Conv1D)	114,816	(None, 161, 128)
batch_normalization_1	512	(Batch (None, 161, 128)
average_pooling1d_2	0	(Average (None, 80, 128)
global_average_pooling1d_1	0	(None, 128)
dense_1 (Dense)	33,024	(None, 256)
batch_normalization_2	1024	(Batch (None, 256)
dense_2 (Dense)	32,896	(None, 128)
BILSTM (Bidirectional LSTM)	1,97,632	(None, 31, 256)
Flatten	0	(None, 7936)
batch_normalization_3	0	(Batch (None, 128)
dense_3 (Dense)	1290	(None, 10)

length vs. Sequences, and 5(c) presents the Protein sequence frequency count.

Experimental results

The protein classes are grouped into four categories: Hydrolase (0), Oxidoreductase (1), Ribosome (2), and Transferase (3).

Table 6 Experimental results for CNN (Base Line Model)

Parameters	Precision %	Recall %	F1-score %	Accuracy %
Hydrolase (0)	82.013	87.236	86.761	86.286
Oxidoreductase (1)	90.631	88.104	84.039	85.603
Ribosome (2)	89.856	89.952	83.791	86.786
Transferase (3)	89.163	87.459	83.014	87.492

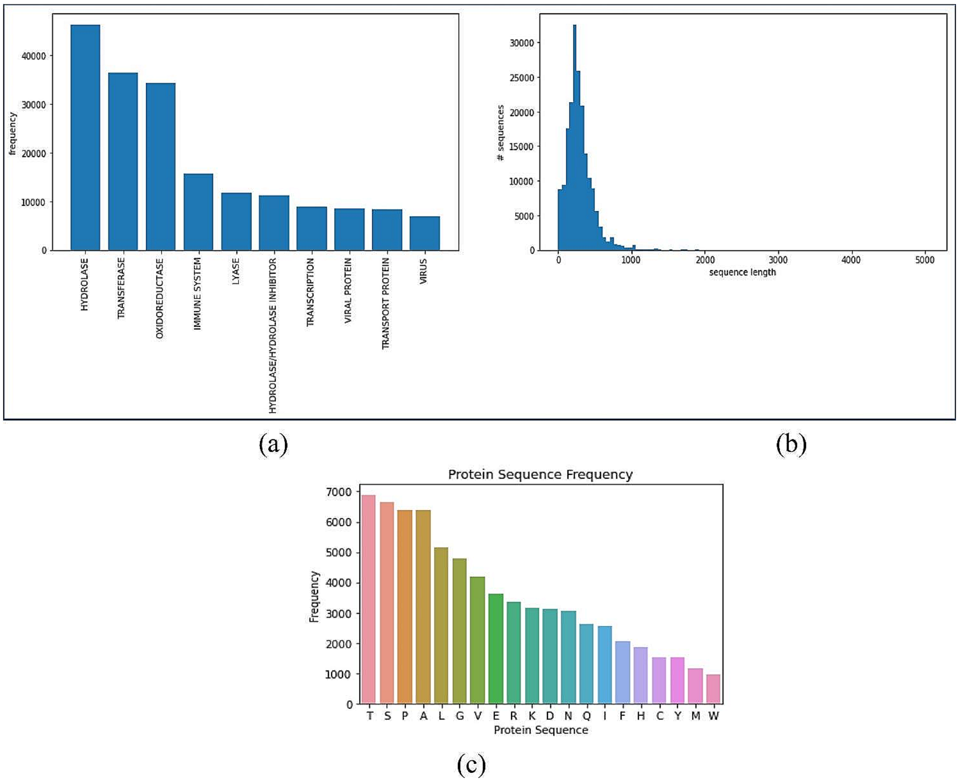
Table 7 Experimental results for CNN-LSTM

Parameters	Precision %	Recall %	F1-score %	Accuracy %
Hydrolase (0)	87.949	88.659	89.042	90.787
Oxidoreductase (1)	87.298	89.476	86.872	89.321
Ribosome (2)	91.791	90.856	92.963	88.709
Transferase (3)	88.486	86.870	87.365	89.326

Table 8 Experimental results for the proposed hybrid model

Parameters	Precision %	Recall %	F1-score %	Accuracy %
Hydrolase (0)	95.371	96.375	94.874	95.074
Oxidoreductase (1)	96.908	97.603	93.271	94.387
Ribosome (2)	95.772	94.667	95.337	96.009
Transferase (3)	93.474	95.187	96.375	97.341

Table 6 describes the experimental results of the CNN model for predicted and actual protein sequence analysis. CNN Model achieved a precision of 82.013% for Hydrolase (0), 90.631% for Oxidoreductase (1), 89.856% for Ribosome (2) and 89.163% for Transferase (3), CNN achieved a recall of 87.236% for Hydrolase (0), 88.104%



(a)

(b)

(c)

Fig. 5 (**a**) Protein sequence frequency of attributes and (**b**) Protein Sequence Length Vs. Sequences and (**c**) Protein sequence frequency count

Table 9 Experimental results comparison of existing vs. proposed models

Technique	Specificity%	Accuracy%	Sensitivity%	MCC%
Existing CNN	85.84	89.27	89.78	81.47
Existing CNN-LSTM	87.37	90.17	88.98	88.35
Proposed Hybrid Model	94.65	96.57	95.67	96.85

for Oxidoreductase (1), 89.952% for Ribosome (2) and 87.459% for Transferase (3).F1-score results of CNN model is 86.761% for Hydrolase (0), 84.039% for Oxidoreductase (1), 83.791% for Ribosome (2) and 83.014% for Transferase (3) and Final results for accuracy is 86.286% for Hydrolase (0), 85.603% for Oxidoreductase (1), 86.786% for Ribosome (2) and 87.492% for Transferase (3).

Table 7 describes the experimental results of the CNN-LSTM model for predicted and actual protein sequence analysis. CNN Model achieved a precision of 87.949% for Hydrolase (0), 87.298% for Oxidoreductase (1), 91.791% for Ribosome (2) and 88.486% for Transferase (3), CNN-LSTM achieved a recall results of 88.659% for Hydrolase (0), 89.476% for Oxidoreductase (1), 90.856% for Ribosome (2) and 86.870% for Transferase (3). F1-score results of CNN model is 89.042% for Hydrolase (0), 86.872% for Oxidoreductase (1), 92.963% for Ribosome (2) and 87.365% for Transferase (3) and Final results for accuracy is 90.787% for Hydrolase (0), 89.321% for Oxidoreductase (1), 88.709% for Ribosome (2) and 89.326% for Transferase (3).

Table 8 describes the experimental results of the Proposed Hybrid model for predicted and actual protein sequence analysis. Proposed Hybrid model achieved a precision of 95.371% for Hydrolase (0), 96.908% for Oxidoreductase (1), 95.772% for Ribosome (2) and 93.474% for Transferase (3), Proposed model achieved a recall results of 96.375% for Hydrolase (0), 97.603% for Oxidoreductase (1), 94.667% for Ribosome (2) and 95.187% for Transferase (3). F1-score results of Proposed model is 94.874% for Hydrolase (0), 93.271% for Oxidoreductase (1), 95.337% for Ribosome (2) and 96.375% for Transferase (3) and Final results for accuracy is 96.074% for Hydrolase (0), 94.387% for Oxidoreductase (1), 96.009% for Ribosome (2) and 97.341% for Transferase (3).

Table 9 presents a comparative analysis of experimental results of existing vs. proposed models. Existing CNN achieved Specificity of 85.84% Accuracy of 89.27%, Sensitivity of 89.78%, and MCC 81.47% and Existing CNN-LSTM achieved Specificity of 87.37%, accuracy of 90.17%, Sensitivity of 88.98%, and MCC 88.35%, and Proposed Hybrid Model achieved Specificity of 94.65%, Accuracy of 96.57%, Sensitivity of 95.67% and MCC 96.85%.

Figure 6 compares the existing CNN, CNN-LSTM, and the proposed model ProtICNN-BiLSTM regarding

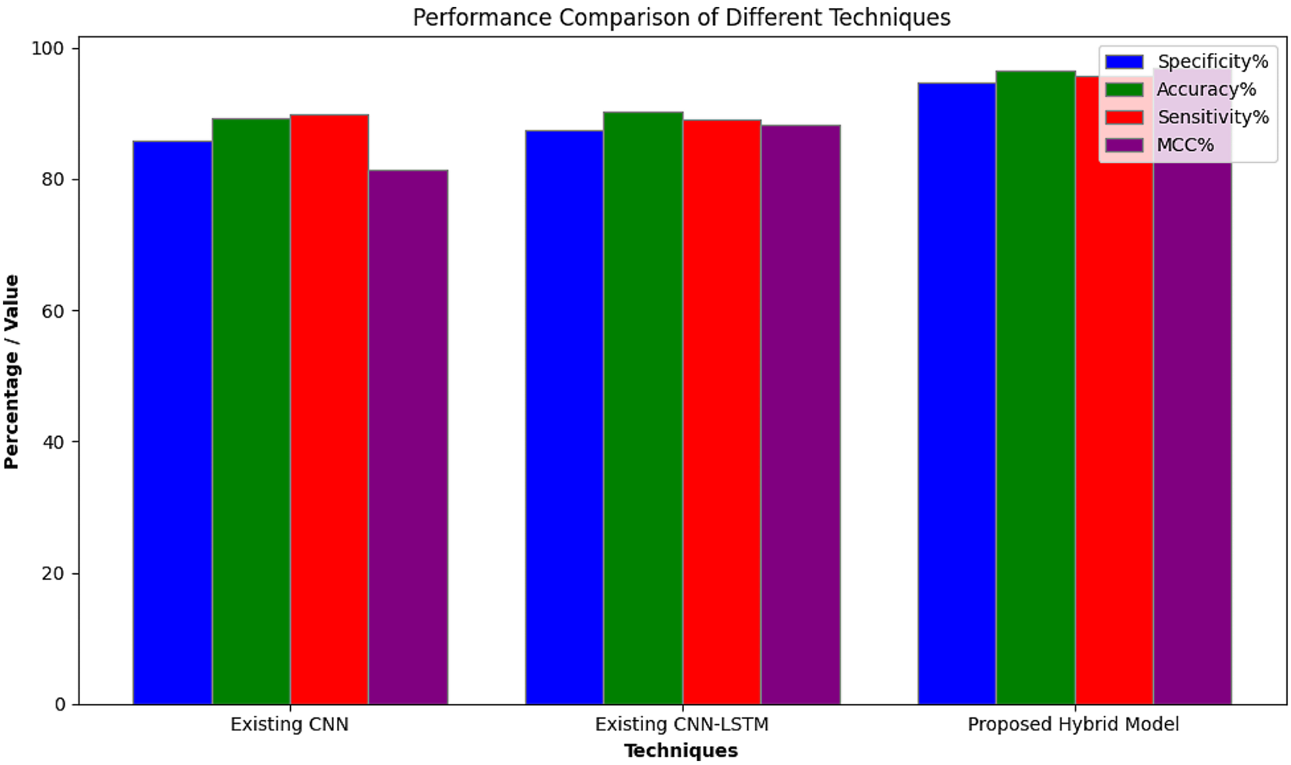


Fig. 6 Comparison of existing and proposed models

Specificity, Accuracy, Sensitivity, and Matthews's correlation coefficient. The proposed model exhibits outstanding performance, achieving higher results for all four parameters. This highlights the proposed model's strength and effectiveness compared to the existing CNN and CNN-LSTM models.

Results and discussion

The fusion of Improved Convolutional Neural Networks and Bidirectional Long Short-Term Memory models, augmented with amino acid embedding techniques, presents a robust strategy for dissecting protein sequences. By harnessing the capabilities of amino acid embedding, the model can effectively exploit the feature extraction prowess inherent in BiLSTM. Subsequent processing by both ICNN and BiLSTM components enables precise prediction of protein attributes, including structural configurations and functional characteristics. However, the efficacy of this approach hinges upon factors such as dataset quality, size, and specific analytical goals.

Experimental assessments were conducted using the PDB-14189 dataset, contrasting the performance of established CNN, CNN-LSTM, and the novel ProtICNN-BiLSTM models. The training spanned 1500 epochs with a batch size 128, optimized via the ADAM optimizer. A meticulous dropout analysis encompassing metrics like Specificity, Sensitivity, Matthews's correlation coefficient, and overall accuracy was undertaken. Results for Different Parameters section delineates binary and multi-class classification outcomes for extant and proposed methodologies. Visualization of protein sequence patterns across diverse classes, encompassing attribute frequencies, sequence length distributions, and sequence counts, is depicted in Fig. 5. The results in Table 6 outline the CNN model's performance, while Table 7 elucidates the CNN-LSTM model's efficacy. Notably, the proposed ProtICNN-BiLSTM model attains remarkable accuracy, recall, F1-score, and support metrics, peaking at 98.11%.

The discernible superiority of the ProtICNN-BiLSTM model over conventional CNN and LSTM variants can be ascribed to several key factors. Firstly, the fusion of ICNN with BiLSTM engenders a holistic approach to capturing local and long-range dependencies within protein sequences. Moreover, incorporating amino acid embedding techniques facilitates a nuanced representation of proteins as numerical vectors, fostering more robust feature extraction. Integrating an attention mechanism further enhances model performance by dynamically weighting the significance of various protein sequence components. Collectively, these advancements underscore the efficacy of the ProtICNN-BiLSTM model in surpassing traditional CNN and LSTM methodologies.

Conclusion and future directions

The Protein-BILSTM model demonstrates the value of collaborative deep learning and molecular biology. The proposed hybrid model surpasses existing approaches with 98.11% accuracy. Hydrolase, Oxidoreductase, Ribosome, and Transferase had similar precision and recall ratings of 87.949–91.791% and 86.870–90.856%. These results prove that the increased biological analysis method works. Though promising, protein sequence categorization needs more research to increase accuracy and application. Protein-BILSTM classifies protein sequences for the first time. CNN and BiLSTM encoding improves accuracy. Its protein structure and activity predictions are accurate enough for sophisticated biological research. This model uses NLP for feature extraction and CNN and BiLSTM for sequential associations. The model's DNA binding prediction summarizes complex biology. The Protein-BILSTM model demonstrates the value of collaborative deep learning and molecular biology. The proposed hybrid model surpasses existing approaches with 98.11% accuracy.

Protein sequence categorization needs more research to improve accuracy and practicality. Research employing more complex datasets can improve model performance and flexibility. Model prediction benefits from protein-protein interaction and functional annotation data. Deep learning and hybrid models can improve efficiency. Protein classifications and databases improve the model. Experimental biologists can verify the model's biological applicability by verifying predictions.

Acknowledgements

This Research is funded by Research Supporting Project Number (RSPD2024R553), King Saud University, Riyadh, Saudi Arabia.

Author contributions

SD & UKL: Contributed to the conceptualization and design of the research project and conducted data collection and analysis. Drafted and revised the manuscript and participated in the development of the research methodology. SS & MA: Contributed to the data collection and statistical analysis and provided critical insights during manuscript drafting. NF & KA: Involved in data acquisition and preprocessing and contributed to the development and implementation of computational models, All the authors Contributed to the theoretical framework and interpretation of results and reviewed and approved the final version of the manuscript.

Funding

This Research is funded by Research Supporting Project Number (RSPD2024R553), King Saud University, and Riyadh, Saudi Arabia.

Data availability

The datasets generated and/or analysed during the current study are available in the PDBj (Protein Data Bank Japan) repository, [https://pdbj.org/emnavi/quick.php?id=EMD-14189]

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 4 May 2024 / Accepted: 7 August 2024
Published online: 27 August 2024

References

1. Yu J, Mu J, Wei T, Hai-Feng Chen. Multi-indicator comparative evaluation for deep learning-based protein sequence design methods. *Bioinformatics* 40, no. 2 (2024): btae037.
2. Zulfiqar H, Guo Z, Ahmad RM, Ahmed Z, Cai P, Chen X, Zhang Y, Lin H. Deep-STP: a deep learning-based approach to predict snake toxin proteins using word embeddings. Front Med. 2024;10:1291352.
3. Ramazi S, Tabatabaei SAH, Khalili E, Nia AG, Motarjem K. Analysis and review of techniques and tools based on machine learning and deep learning for prediction of lysine malonylation sites in protein sequences. *Database* 2024 (2024): baad094.
4. Ghosh S, Mitra P. MaTPIP: a deep-learning architecture with eXplainable AI for sequence-driven, feature mixed protein-protein interaction prediction. Comput Methods Programs Biomed. 2024;244:107955.
5. He J, Wu W, Wang X. DIProT: a deep learning based interactive toolkit for efficient and effective protein design. Synth Syst Biotechnol (2024), 32–86.
6. Tahir M, Khan F, Hayat M, Mohammad Dahman Alshehri. An effective machine learning-based model for the prediction of protein–protein interaction sites in health systems. Neural Comput Appl. 2024;36(1):65–75.
7. Ali S, Sahoo B, Zelikovsky A, Chen P-Y, Patterson M. Benchmarking machine learning robustness in Covid-19 genome sequence classification. Sci Rep. 2023;13(1):4154.
8. Yeung W, Zhou Z, Mathew L, Gravel N, Taujale R, Boyle BO, Salcedo M, et al. Tree visualizations of protein sequence embedding space enable improved functional clustering of diverse protein superfamilies. Brief Bioinform. 2023;24(1):bbac619.
9. Motmaen A, Dauparas J, Baek M, Abedi MH, Baker D, Bradley P. Peptide-binding specificity prediction using fine-tuned protein structure prediction networks. *Proceedings of the National Academy of Sciences* 120, no. 9 (2023): e2216697120.
10. Yao J, Ling Y, Hou P, Wang Z, Huang L. A graph neural network model for deciphering the biological mechanisms of plant electrical signal classification. Appl Soft Comput. 2023;137:110153.
11. Goto K, Tamehiro N, Yoshida T, Hanada H, Sakuma T, Adachi R. Kazunari Kondo, and Ichiro Takeuchi. Novel machine learning method allerStat identifies statistically significant allergen-specific patterns in protein sequences. J Biol Chem. 2023;299:6.
12. Hou Z, Yang Y, Ma Z, Wong Ka-chun, Li X. Learning the protein language of proteome-wide protein-protein binding sites via explainable ensemble deep learning. Commun Biology. 2023;6(1):73.
13. Llinares-López F, Berthet Q, Blondel M, Teboul O. Deep embedding and alignment of protein sequences. Nat Methods. 2023;20(1):104–11.
14. Singh D, Roy J. A large-scale benchmark study of tools for the classification of protein-coding and non-coding RNAs. Nucleic Acids Res. 2022;50(21):12094–111.
15. Tripathi R, Patel S, Kumari V, Chakraborty P, Pritish Kumar V. DeepLNC, a long non-coding RNA prediction tool using deep neural network. Netw Model Anal Health Inf Bioinf. 2016;5:1–14.
16. Hashem, Abu MAM, Hossain AR, Marlinda MA, Mamun S, Sagadevan Z, Shahnavaz. Khanom Simarani, and Mohd Rafie Johan. Nucleic acid-based electrochemical biosensors for rapid clinical diagnosis: advances, challenges, and opportunities. Crit Rev Clin Lab Sci. 2022;59(3):156–77.
17. Erten M, Aydemir E, Barua PD, Baygin M, Dogan S, Tuncer T, Tan R-S. Abdul Hafeez-Baig, and U. Rajendra Acharya. Novel tiny textural motif pattern-based RNA virus protein sequence classification model. Expert Syst Appl. 2024;242:122781.
18. Ahmed N, Yousif WA, Alsanousi EM, Hamid MK, Elbashir KM, Al-Aidarous. Mogtaba Mohammed, and Mohamed Elhafiz M. Musa. An efficient Deep Learning Approach for DNA-Binding proteins classification from primary sequences. Int J Comput Intell Syst. 2024;17(1):1–14.
19. Onyema EM, Lilhore UK, Saurabh P, Dalal S, Nwaeze AS. Asogwa Tochukwu Chijindu, Lauritta Chinazaekpere Ndufeiya-Kumasi, and Sarita Simaiya. Evaluation of IoT-Enabled hybrid model for genome sequence analysis of patients in healthcare 4.0. Measurement: Sens. 2023;26:100679.
20. Pattnaik D, Thakur SB, Dash PM, Jena S. Sumanta Sahu, and Sulin Kumar Behera. Molecular Medical diagnosis of COVID-19 and Omicron variant. J Pharm Negat Results (2022): 6332–47.
21. Chhabra C. and Meghna Sharma. Machine Learning, Deep Learning and Image Processing for Healthcare: A Crux for Detection and Prediction of Disease. In *Proceedings of Data Analytics and Management: ICDAM 2021, Volume 2*, pp. 305–325. Springer Singapore, 2022.
22. Le NQ, Khanh Q-T, Ho T-T-D, Nguyen, Yu-Yen Ou. A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. Brief Bioinform. 2021;22(5):bbab005.
23. Tavakoli N. Seq2image: Sequence analysis using visualization and deep convolutional neural network. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1332–1337. IEEE, 2020.
24. Ho Q-T, Yu-Yen Ou. Classifying the molecular functions of Rab GTPases in membrane trafficking using deep convolutional neural networks. *Analytical biochemistry* 555 (2018): 33–41.
25. Sureyya Rifaioglu, Ahmet T, Doğan MJ, Martin. Rengul Cetin-Atalay, and Volkan Atalay. DEEPred: automated protein function prediction with multi-task feed-forward deep neural networks. *Scientific reports* 9, no. 1 (2019): 1–16.
26. Liu C-M, Ta V-D. Nguyen Quoc Khanh Le, Direselign Addis Tadesse, and Chongyang Shi. Deep neural network framework based on word embedding for protein Glutarylation sites prediction. *Life* 12, no. 8 (2022): 1213.
27. Yuvaraj N, Srihari K, Chandragandhi S, Raja RA, Dhiman G, Kaur A. Analysis of protein-ligand interactions of SARS-Cov-2 against selective drug using deep neural networks. Big Data Min Analytics. 2021;4(2):76–83.
28. Le NQ, Khanh EKY, Yapp Y-Y, Ou, Hui-Yuan Yeh. iMotor-CNN: Identifying molecular functions of cytoskeleton motor proteins using 2D convolutional neural network via Chou's 5-step rule. *Analytical biochemistry* 575 (2019): 17–26.
29. Pu L, Govindaraj RG, Lemoine JM, Wu H-C, Brylinski M. DeepDrug3D: classification of ligand-binding pockets in proteins with a convolutional neural network. PLoS Comput Biol. 2019;15(2):e1006718.
30. Zhang Z, Park CY, Theesfeld CL, Olga G. Troyanskaya. An automated framework for efficiently designing deep convolutional neural networks in genomics. Nat Mach Intell. 2021;3(5):392–400.
31. Wang Y-B, You Z-H, Li X, Jiang T-H, Chen X. Xi Zhou, and Lei Wang. Predicting protein–protein interactions from protein sequences by a stacked sparse autoencoder deep neural network. Mol Biosyst. 2017;13(7):1336–44.
32. Xu M, Papageorgiou DP, Sabia Z, Abidi M, Dao. Hong Zhao, and George Em Karniadakis. A deep convolutional neural network for classification of red blood cells in sickle cell anemia. PLoS Comput Biol. 2017;13(10):e1005746.
33. Niu M, Lin Y, Zou Q. sgRNACNN: identifying sgRNA on-target activity in four crops using ensembles of convolutional neural networks. Plant Mol Biol. 2021;105:483–95.
34. Zhao T, Hu Y, Valsdottir LR. Tianyi Zang, and Jiajie Peng. Identifying drug–target interactions based on graph convolutional network and deep neural network. Brief Bioinform. 2021;22(2):2141–50.
35. Deng L, Liu Y, Shi Y, Zhang W, Yang C, Liu H. Deep neural networks for inferring binding sites of RNA-binding proteins by using distributed representations of RNA primary sequence and secondary structure. *BMC genomics* 21, no. 13 (2020): 1–10.
36. Wang L, Wang H-F, Liu S-R, Xin Y, Ke-Jian S. Predicting protein-protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. Sci Rep. 2019;9(1):9848.
37. Ben-Bassat I, Chor B, Orenstein Y. A deep neural network approach for learning intrinsic protein-RNA binding preferences. Bioinformatics. 2018;34:17.
38. Le NQ, Khanh, Quang-Thai H. Deep transformers and convolutional neural network in identifying DNA N6-methyladenine sites in cross-species genomes. Methods. 2022;204:199–206.
39. Mitra S, Saha S, Hasanuzzaman M. A multi-view deep neural network model for chemical-disease relation extraction from imbalanced datasets. IEEE J Biomedical Health Inf. 2020;24(11):3315–25.
40. Lv Z, Ding H, Wang L, Zou Q. A convolutional neural network using dinucleotide one-hot encoder for identifying DNA N6-methyladenine sites in the rice genome. Neurocomputing. 2021;422:214–21.
41. Taju S, Wellem T-T-D, Nguyen N-Q-K, Le. DeepEfflux: a 2D convolutional neural network model for identifying families of efflux proteins in transporters.

Bioinformatics. 2018;34(18):3111–7. Rosdyana Mangir Irawan Kusuma, and Yu-Yen Ou.

42. Zhang Y, Qiao S, Ji S, Li Y. DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding. Int J Mach Learn Cybernet. 2020;11:841–51.
43. Cheng J, Liu Y, Ma Y. Protein secondary structure prediction based on integration of CNN and LSTM model. J Vis Commun Image Represent. 2020;71:102844.
44. Pang L, Wang J, Zhao L, Wang C, Zhan H. A novel protein subcellular localization method with CNN-XGBoost model for Alzheimer's disease. Front Genet. 2019;9:751.
45. Park S. GalaxyWater-CNN: prediction of water positions on the protein structure by a 3D-convolutional neural network. J Chem Inf Model. 2022;62(13):3157–68.
46. Zhang D, Mansur R. Kabuka. Protein family classification from scratch: a CNN based deep learning approach. IEEE/ACM transactions on computational biology and bioinformatics 18, 5 (2020): 1996–2007.
47. Peng Y, Rios A, Ramakanth Kavuluru, and, Lu Z. Chemical-protein relation extraction with ensembles of SVM, CNN, and RNN models. arXiv preprint arXiv:1802.01255 (2018).
48. Barukab O, Ali F, Alghamdi W, Bassam Y, Sher Afzal Khan. DBP-CNN: deep learning-based prediction of DNA-binding proteins by coupling discrete cosine transform with two-dimensional convolutional neural network. Expert Syst Appl. 2022;197:116729.
49. Jiang M, Wei Z, Zhang S, Wang S, Wang X, Li Z. Frsite: protein drug binding site prediction based on faster R–CNN. J Mol Graph Model. 2019;93:107454.
50. Lin Z, Lanchantin J, Qi Y. MUST-CNN: a multilayer shift-and-stitch deep convolutional architecture for sequence-based protein structure prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1. 2016.

Publisher's Note