International Journal of Computational Intelligence Systems (2024) 17:88
https://doi.org/10.1007/s44196-024-00462-3

RESEARCH ARTICLE

An Efficient Deep Learning Approach for DNA-Binding Proteins Classification from Primary Sequences

Nosiba Yousif Ahmed[1,2] · Wafa Alameen Alsanousi[1,3] · Eman Mohammed Hamid[1] · Murtada K. Elbashir[4] · Khadija Mohammed Al-Aidarous[2] · Mogtaba Mohammed[5] · Mohamed Elhafiz M. Musa[6]

Received: 14 December 2023 / Accepted: 17 March 2024

Abstract

As the number of identified proteins has expanded, the accurate identification of proteins has become a significant challenge in the field of biology. Various computational methods, such as Support Vector Machine (SVM), K-nearest neighbors (KNN), and convolutional neural network (CNN), have been proposed to recognize deoxyribonucleic acid (DNA)-binding proteins solely based on amino acid sequences. However, these methods do not consider the contextual information within amino acid sequences, limiting their ability to adequately capture sequence features. In this study, we propose a novel approach to identify DNA-binding proteins by integrating a CNN with bidirectional long-short-term memory (LSTM) and gated recurrent unit (GRU) as (CNN-BiLG). The CNN-BiLG model can explore the potential contextual relationships of amino acid sequences and obtain more features than traditional models. Our experimental results demonstrate a validation set prediction accuracy of 94% for the proposed CNN-BiLG, surpassing the accuracy of machine learning models and deep learning models. Furthermore, our model is both effective and efficient, exhibiting commendable classification accuracy based on comparative analysis.

Keywords Deep learning · Convolutional neural network · Gated recurrent unit · Long-short term memory · DNA-binding protein · Protein classification

1 Introduction

The family of macromolecules called DNA-binding proteins (DBPs) are essential for many biological functions, including gene control, DNA replication, repair, and recombination [1, 2]. Their importance includes essential biological processes such as alternative splicing, methylation, and RNA editing [3]. Understanding the relationships between DBPs and DNA becomes essential given their crucial function in biological processes [4]. Interestingly, DBP also has a significant impact on human health. Some variations have been linked to cancer and other chronic diseases, while others have contributed to the design of drugs, including steroids, anti-inflammatories, and antibiotics [5]. Recent studies showing that more than 3% of eukaryotic and prokaryotic proteins can bind DNA highlights the existence of DBP-DNA interactions and the ubiquity of DBPs in biological systems [6, 7]. However, many obstacles stand in the way of identifying and characterizing DBP. Conventional experimental techniques, such as X-ray crystallography and filter binding assays, can be expensive and time-consuming [8].

✉ Khadija Mohammed Al-Aidarous
bintalameen@gmail.com

1 Department of Computer Science, Faculty of Mathematical and Computer Science, University of Gezira, Wad Madani, Sudan

2 Department of Computer Science, College of Science and Arts, Najran University, Sharorah, Saudi Arabia

3 Alghad College for Applied Medical Sciences, Riyadh, Saudi Arabia

4 Department of Information Systems, College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia

5 Department of Mathematics, Faculty of Sciences AL-Zulfi, Majmaah University, 11952 Al Majmaah, Saudi Arabia

6 Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia

Published online: 11 April 2024

On the other hand, computational methods offer a viable route to rapid and cost-effective detection of BPD [1].

The categorization and identification of DNA-binding proteins is critical to understanding several biological processes, such as transcriptional control, DNA repair, and gene regulation [9, 10]. Conventional protein classification methods often rely on human-like feature creation and shallow learning strategies, which may not be able to fully capture the complex relationships and patterns observed in protein sequences [11, 12]. Recent developments in deep learning have shown the potential to address this difficulty by enabling the direct extraction of meaningful representations from raw sequence data, thereby leading to more accurate and efficient categorization [13]. For example, the work of Koo and Ploenzke [14] illustrated the promise of these approaches for understanding complex biological processes by demonstrating the effectiveness of deep learning models in predicting the DNA sequence specificity of transcription factors. Computational methods using machine learning (ML) and deep learning (DL) techniques have the potential to change the categorization of DBPs by providing rapid and accurate predictions [15–17]. The rapid advancement progress of DL made in the early 2000s is well positioned to meet the challenges of bioinformatics, particularly using the enormous potential of biological big data [18]. Convolutional neural networks (CNN) have been an effective tool in this context, particularly in the field of genomics research [19]. By processing genomic data as fixed-length 1D sequences, CNNs can be adapted to perform tasks such as occupancy prediction and motif identification [20].

Many computational methods have been developed to discover DNA-binding proteins (DBPs) from base sequences, but each presents its difficulties [21, 22]. Key phases of these strategies include creating effective feature sets and selecting appropriate machine learning algorithms [23]. For DBP prediction, conventional machine learning models such as Support Vector Machine (SVM) and Random Forest (RF) have been widely used. For example, Jia et al. successfully integrated the features of position-specific scoring matrices (PSSM) with RF to create the KK-DBP method, which achieved a success rate of 81.22% Jia et al. [24]. SVM with multiple kernel learning was used by Qian et al. [25] to outperform previous techniques on benchmark datasets. To improve the accuracy of DBP predictions, Sang et al. [26] and Wang et al. [27] respectively, adopted SVM and Hidden Markov Model (HMM) profiles. Similarly, Ma et al. [28] presented the DNABP method for DBP detection, which combines RF classifiers and hybrid features. In addition, several sequence-based methods and web servers, such as (MK-FSVM-SVDD) [29], (DBPPred-PDSD) [30], (MSFBinder) [31] (Local-DPP) [32], (HMMBinder) [33], and (SVM-PSSM-DT) [34], have been developed for identification of DBPs. On the other hand, huge datasets are a limitation for classical ML algorithms, and feature extraction, training, and prediction require specialized knowledge [35].

DL has recently been successfully used for a variety of massive dataset categorization challenges [36]. When computing vast amounts of DNA sequence data, DL technology offers incomparable benefits. For example [35] introduced a deep learning model named KEGRU, a model that merges the Bidirectional GRU network with k-mer embedding to detect TF binding sites. Researchers [37] predicted DBPs from primary protein sequences by comparing the accuracy of the model in a DL-based procedure and counting prediction analysis of precision, recall, f-measure, and false discovery rate of the protein sequence. Zhang et al. [38] have introduced a novel predictor, coined as ENSEMBLE-CNN. This predictor amalgamates instance selection and bootstrapping methodologies to forecast imbalanced DNA-binding sites from protein primary sequences. Moreover, ENSEMBLE-CNN has attained exceptional prediction accuracy and has surpassed the performance of currently existing sequence-based protein-DNA binding site predictors. Correspondingly, the researcher [39] used artificial Recurrent Neural Networks (RNNs) for the direct classification of protein function based solely on primary sequence, without the need for sequence alignment, heuristic scoring, or feature engineering. A DL neural network for DNA sequence classification based on spectral sequence representation is presented by [40]. This demonstrated that the DL approach outperformed all the other classifiers when considering the classification of small sequence fragments 500 bp long. The researchers [41] started their study by examining the prior classification approaches, namely alignment methods, and highlighting their limitations. They subsequently delve into the realm of DL, encompassing artificial neural networks and hyperparameter tuning. Finally, they showcase the latest state-of-the-art DL architectures utilized in the classification of DNA. Furthermore, the researcher [42] presented two distinct DL approaches, namely DeepDBP-ANN and DeepDBP-CNN, for the detection of DBPs. These methods have demonstrated exceptional performance on standard benchmark datasets, thereby establishing new benchmarks for this task.

The Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM) model acquires more features than conventional models and examines the potential contextual correlations of amino acid sequences [43]. The earlier researcher [1] presented a DL technique for identifying DBPs using CNN and Long Short-Term Memory (LSTM) neural networks with binary cross-entropy for network quality assessment. The earlier researchers [38] developed a two-level predictor called DeepDRBP-2L by fusing the LSTM and CNN to identify DBPs, and RBPs. The researchers [44] presented a novel framework called MPPIF-Net that utilized DL with multilayer bi-directional LSTM to accurately identify

Plasmodium falciparum parasite mitochondrial proteins, outperforming existing approaches. The researcher [45] introduced the PDBP-Fusion method, which utilized DL techniques to predict DBPs by incorporating local features and long-term dependencies from primary sequences with a Bi-LSTM network and a CNN. They apply the method on the PDB2272 independent dataset and an online server to improve DBP prediction. The researcher [46] used a transfer learning method to transfer samples and build data sets, where two features were retrieved from a protein sequence and two traditional transfer learning methods were compared. The last phase involved creating a DL neural network model that took advantage of attention mechanisms to find DBPs. The researcher [35] proposed a hybrid deep learning framework called DeepD2V for predicting transcription factor binding sites from DNA sequences. The method combines a sliding window method with word2vec-based k-mer distributed representation, recurrent neural networks, and convolutional neural networks. To categorize the transcription factor proteins of primates, researchers [47] suggested a deep learning model that combines a Word2Vec preprocessing step with a hybrid structure of RNN-based LSTM and GRU networks.

Existing methods for classifying DNA-binding proteins are limited because they cannot extract features from amino acid sequences and ignore contextual information [48]. These methods often overlook important patterns suggesting DNA binding properties by failing to capture contextual interactions between amino acids. Additionally, the complex nature of DNA–protein interactions poses a challenge, as available methods may not fully capture the range of structural and functional properties displayed by DNA-binding proteins. Furthermore, there is a pressing demand to improve the prediction performance in this identification process because DNA-binding proteins are important for various applications in molecular biology and bioinformatics [49]. These challenges are addressed by the proposed method, which combines convolutional neural networks (CNN) with bidirectional long short-term memory (LSTM) and gated recurrent unit (GRU) layers [50]. This allows contextual dependencies within amino acid sequences to be captured and potential interactions between amino acids to be explored. Experimental results indicate that this improves feature extraction and increases prediction accuracy.

Amidst the tricky challenges and exciting opportunities that the study of DNA-binding proteins (DBPs) presents, our goal is simple: to create a new computational method that eliminates the drawbacks of old-fashioned experiments. We are developing a new method that mixes different types of smart layers with neural networks. By paying attention to the small details of amino acid sequences, we hope to make better guesses about proteins and extract properties more accurately. Our goal is to prove that this new method works well by testing it extensively, which could advance our understanding of biology and bioinformatics. We built a special DBP sorting framework, and we refined it to be extremely accurate and efficient. Our results show that this method is very powerful, as it helps us discover hidden patterns in large data sets. Ultimately, our project is not only about solving today's problems, it is also about opening doors to great discoveries in biology and bioinformatics.

2 Materials and Methods

We categorize DNA binding proteins (DBPs) using a novel CNN-BiLG architecture. Data quality and consistency are ensured by careful data collection, which involves obtaining a diverse dataset of DBP-related protein sequences from reliable sources and then going through rigorous preprocessing steps such as sequence alignment and deduplication. We then present the CNN-BiLG architecture, a state-of-the-art neural network framework that efficiently captures local and global features of protein sequences by combining CNNs and bidirectional long-term memory layers. For a deep understanding of the temporal properties of data, CNN layers use convolutional filters to extract local features, while BiLSTM layers record sequential dependencies within sequences. The model's ability to detect meaningful patterns in input sequences is improved using feature fusion methods to combine features generated by the CNN and BiLSTM layers. Predictions for protein classification are generated by the classification head, which uses the fused features and is composed of fully linked layers with softmax activation. We use cross-validation methods to measure the robustness and generalizability of the proposed framework, and we calculate conventional metrics such as recall, accuracy, precision, and F1 score to measure its performance. Furthermore, to prove its effectiveness and excellence, the CNN-BiLG architecture is tested in protein classification tasks against baseline models and state-of-the-art methods. The CNN-BiLG architecture is visually represented in the following schematic diagram (Fig. 1), which shows the flow of information through the many levels and components of the framework. We hope that by providing this comprehensive technology, we can help improve computational tools for biological research and drug discovery by improving protein categorization.

2.1 Data Acquisition and Preparation

The dataset has been taken from [1], which extracted protein sequences from the Swiss-Prot dataset [51], a widely recognized database of protein sequences and associated functional information. Specifically, the raw dataset was derived from the 2016.5 release version of Swiss-Prot and comprised 551,193 proteins. To isolate DBPs, the authors

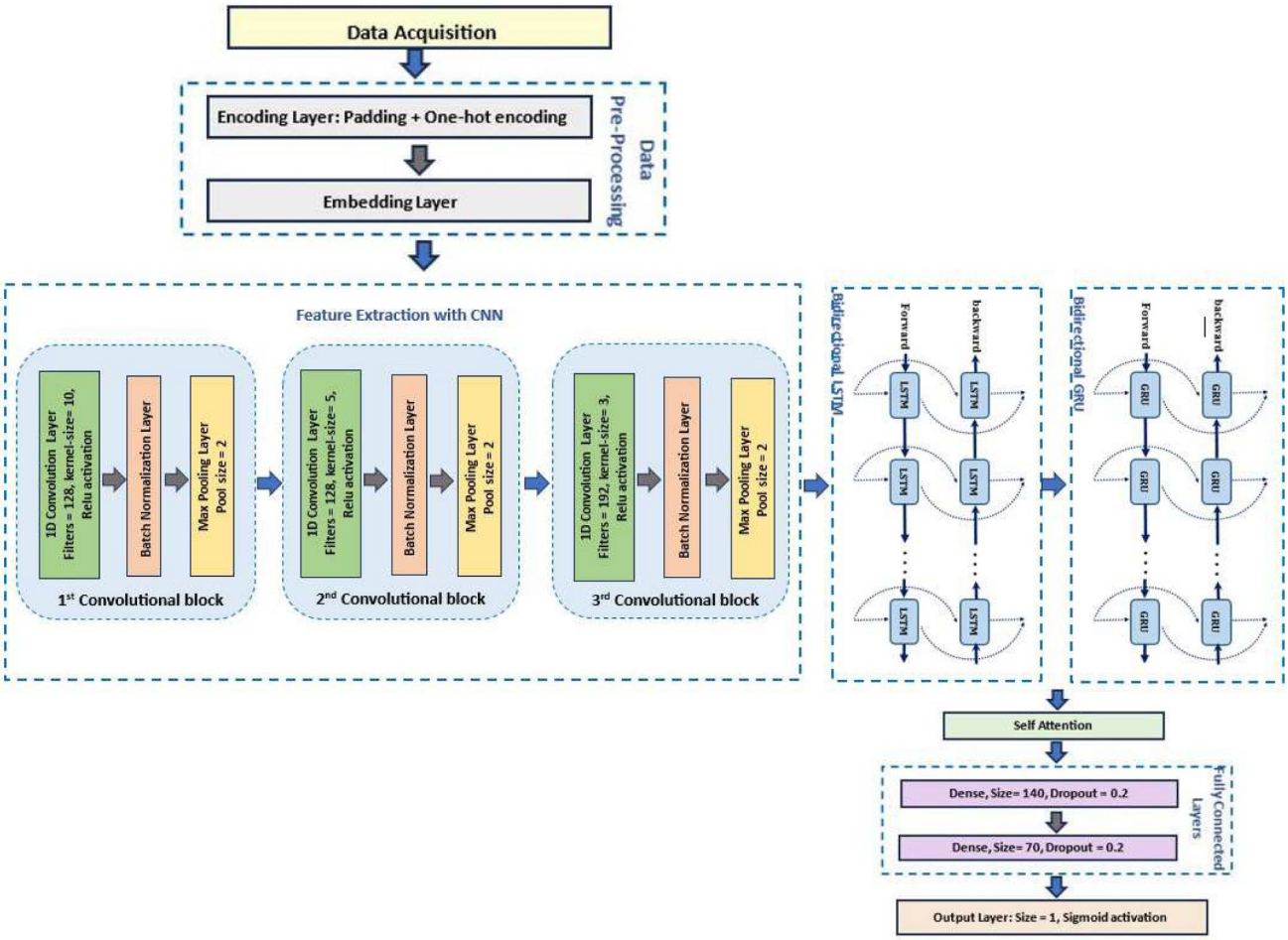

Fig. 1 Overview of the proposed hybrid architecture for DNA binding proteins classification

Table 1 Optimized dataset splitting strategy for protein sequence classification, symmetry between DNA-binding and non-DNA-binding samples

Data set	DNA-binding	Non-DNA-binding	Total
Original set	42,257	42,310	84,567
Train set	33,805	33,848	67,653
Test set	8452	8462	16,914

[1] conducted a keyword search for sequences containing the term "DNA-Binding" and applied a size filter to remove those with either a length less than 40 or greater than 1000 amino acids. Ultimately, a collection of 42,257 protein sequences was identified as positive samples. To generate negative samples, 42,310 non-DBPs were randomly selected from the remaining dataset using the query condition molecule function and length. The positive and negative samples were subsequently divided into training and testing sets, with 80% of the data assigned for training purposes and the remainder used for testing, as listed in Table 1.

Notably, conventional sequence-based classification methods frequently encounter the issue of over-fitting due to the existence of redundancy in the training dataset, leading to inflated performance metrics. To tackle this problem, the authors [1] utilized the CD-HIT tool with a threshold value of 0.7 to remove sequence redundancy.

2.2 Data Pre-processing

In DL models, all input and output variables must be numerical; hence, before model fitting and evaluation, data must be converted from categorical to numerical format. The two most prevalent techniques for encoding categorical variables are one-hot encoding and ordinal encoding. In the proposed architecture, we implemented the one-hot encoding technique, in which binary vectors represent the categorical variables. To accomplish this, the categorical values must first undergo conversion into integer numbers. The index of the

integer, denoted with a 1, is used to depict each integer value as a binary vector, with all other values represented as zero. It is noteworthy that the outcome is not influenced by the protein sequence encoding, although assigning a regular number to each amino acid results in the encoding technique creating a digital vector of a protein sequence with a predetermined length, as shown in Table 2.

The vector space model is a critical concept in Natural Language Processing (NLP) that enables the representation of words in a continuous vector space. The embedding technique is commonly employed to map semantically related phrases to semantically related places in vector space. A weight matrix is added to the one-hot vector from the left to achieve this, with the weight matrix Wϵ $R^{d\ X|V|}$ dimension being determined by Eq. 1, which accounts for the number of distinct symbols in the lexicon represented by |V|. The output of the embedding layer is a series of dense real-valued vectors such as (V_1, V_2... V_n), each having a fixed vector length. The output vector length in the embedding layer is 8×1, and the sequence is transformed into an 8×8 matrix due to layer proteins [52].

$$V_n = W_{X_n} \tag{1}$$

Table 2 Encoding amino acids: transforming categorical variables into digital vectors for predetermined length protein sequences

Amino acids	Letters	Code
Alanine	A	1
Cysteine	C	2
Aspartic	D	3
Glutamic	E	4
Phenylalanine	F	5
Glycine	G	6
Histidine	H	7
Isoleucine	I	8
Lysine	K	9
Leucine	L	10
Methionine	M	11
Asparagine	N	12
Proline	P	13
Glutamine	Q	14
Arginine	R	15
Serine	S	16
Threonine	T	17
Valine	V	18
Tryptophan	W	19
Tyrosine	Y	20
Illegal Amino acids	B, U, J, Z, O	22, 23, 24, 25, 26

2.3 Feature Extraction via CNN

In this study, we utilize the CNN algorithm of DL to extract hidden useful information from proteins. The CNN is a feed-forward neural network comprising neurons that respond to the surrounding units in a part of the coverage, making it an excellent performer for data feature extraction. The CNN operates using forward propagation to calculate the output value and backpropagation to adjust the weights and biases. It is composed of five layers, namely, the input layer, convolution layer, pooling layer, full connection layer, and output layer [53].

The input layer of the CNN is responsible for receiving the data, while the output layer is responsible for producing the final output of the network. The convolution layer of the CNN is responsible for identifying patterns in the data. By applying filters to the input data, the convolution layer can detect features that are relevant to solving the problem at hand. The pooling layer of the CNN is responsible for reducing the dimensionality of the data. By removing extraneous information from the data, the pooling layer can reduce the size of the network, improving its performance. The fully connected layer is essentially fed forward neural networks that compose the network's last few levels [41].

In this study, convolutional neural networks can process the encoded amino acid sequence since it was transformed into a fixed-size two-dimensional matrix as it traveled through the embedding layer. The proposed CNN comprises three 1-D convolution layers and three max-pooling layers, which serve as non-linear activation layers to decrease the feature map size. More details about the parameters of each layer are given in Table 3.

2.4 Long Short-Term Memory (LSTM) and Bidirectional LSTM (biLSTM)

LSTM is a type of RNN. The addition of "gates" by LSTM allows it to filter out memory regions that are unimportant to prediction and to regulate the degree of influence of data from the previous and current stages. More flexible memory control is possible because of this method [38].

The internal structure review of LSTM contains the memory ell, which directly relates to (H_{st-1}), the time, and the successive state (X_{st}) h controls the internal state v or hatted to be upgraded. There are three gates in the LST subscripts structure input gates (N_{st}) forget gates (F_{st}) and output gate ($O_{st)}$ as shown in Fig. 2a. The mathematical notation of these gates is as follows:

$$\text{e}\,N_{st} = \sigma\left(W_n\left[H_{st-1}, X_{st}\right] + b_n\right) \tag{2}$$

$$\text{Forget gate}\,F_{st} = \sigma\left(W_f\left[H_{st-1}, X_{st}\right] + b_f\right) \tag{3}$$

Table 3 Enhancing the model understanding for layers, parameters, and output shapes for complete model evaluation and validation

Layer (type)	Parameters	Output shape
Input_1(InputLayer)	Sentence_length = 1000	(None, 1000)
	n_batches = 64	
Embedding (Embedding)	Input_dim = 2944	(None, 1000, 128)
	Embedding_size = 128	
Spatial_dropout1d ((Spatial Dropout1D)	–	(None, 1000, 128)
Conv1d (Conv1D)	Learnable weights and biases = 163,968	(None, 991, 128)
	Output_dim = 128	
	Filter_length = 10	
	Activation = relu	
Batch_normalization (Batch Normalization)	Trainable parameters = 512	(None, 991, 128)
Max_pooling1d (MaxPooling 1D)	Pool_length = 2	(None, 245, 128)
Conv1d_1 (Conv1D)	Learnable weights and biases = 82,048	(None, 491, 128)
	Output_dim = 128	
	Filter_length = 5	
	activation = relu	
Batch_normalization_1 (Batch Normalization)	TRAINABLE parameters = 512	(None, 491, 128)
Max_pooling1d_1 (MaxPooling 1D)	Pooling_length = 2	(None, 245, 128)
Conv1d_2 (Conv1D)	Learnable weights and biases = 73,920	(None, 243, 192)
	Output_dim = 192	
	Filter_length = 3	
	Activation = relu	
Batch_normalization_2 (Batch Normalization)	Trainable parameters = 768	(None, 243, 192)
Max_pooling1d_2 (MaxPooling 1D)	Pooling_length = 2	(None, 121, 192)
Bidirectional ((Bidirectional)	Learnable weights and biases = 147,280	(None, 121, 140)
	Lstm_output_size = 70	
Bidirectional_1((Bidirectional)	Learnable weights and biases = 89,040	(None, 121, 140)
	Lstm_output_size = 70	
Attention (Attention)	Attention heads = 1	(None, 121, 140)
Dense (Dense)	Learnable weights and biases = 19,740	(None, 121, 140)
Dropout (Dropout)	0	(None, 121, 140)
Dense_1 (Dense)	Learnable weights and biases = 9870	
Global_average_pooling1d (GlobalAveragePooling1D)		(None, 70)
Dense_2 (Dense)	Learnable weights and biases = 71	(None, 1)
Dense_3 (Dense)	Learnable weights and biases = 2	(None, 1)

$$\text{Output gate } O_{st} = \sigma\left(W_0\left[H_{st-1}, X_{st}\right] + b_0\right) \tag{4}$$

where (W_n, W_f, W_o) are the weight matrices for the input, forget, and output gates, respectively. (b_n, b_f), and (b_o) e the bias vectors for the corresponding gates, and σ is the sigmoid function. The input gate (N_{st}) determines which values to update, while the forget gate (F_{st}) determines which values to forget. The output gate (O_{st}) determines which values to output from the memory cell. LSTM provides a practical solution to the difficulties encountered in RNNs, particularly in the storage and processing of long sequences. Using memory cells and gates, LSTM can effectively learn and store information while avoiding the vanishing gradient problem. An addition to LSTM called BiLSTM starts from the final timestep of the forward recurrence and moves backward to the first timestep of the forward recurrence.

Thus, it is possible to record the knowledge in the "future" stages and use it to support predictions at earlier time steps [37]. In the proposed method, we use BiLSTM with a dropout of 0.3 to reduce the gap between training and validation.

2.5 Gated Recurrent Unit (GRU) and Bidirectional GRU (BiGRU)

The GRU is a type of sequential model specifically designed to tackle the issue of long-term dependencies. These dependencies can lead to the problem of vanishing gradients in larger, more traditional neural networks. The issue is resolved by the retention of previous time point memory to enhance the network's ability to make more accurate predictions in the future. The construction of gates is a focal point for GRU, as they regulate information processing and storage and enable the network's hidden states to be modified and disregarded. The update gate in GRU's internal structure review decides what data to discard and what new material to add, while the reset gate decides how much past knowledge to remove [35]. In Fig. 2b, the update gate (z_t), reset gate (r_t), applicant hidden state ($h_t^{\sim}$) of the presently hidden node (h_t) current hidden state (h_t), current neural network input (x_t), and previously hidden state h_{t-1}) are all denoted. The whole set of calculation Eqs. (5–8) is shown below. The sigmoid activation function ranges from 0 to 1. It assesses the value of earlier data before applying it to the candidate for the current value. The matrix's Hadamard product, shown by the circular dot ($\odot$), produces an output between 0 and 1. Filtering the prior cell state (h_{t-1}) and the updated candidate ($h_t^{\sim}$) provides the current cell state (h_t). To compute the current cell state and the amount of prior cell state that is kept, the update gate (z_t) specifies the number of updated candidates that are needed [54]. The architecture of LSTM and GRU can be represented in Fig. 2a and b.

$$z_t = \sigma(w_{zx}x_t + u_{zh}h_{t-1}) \tag{5}$$

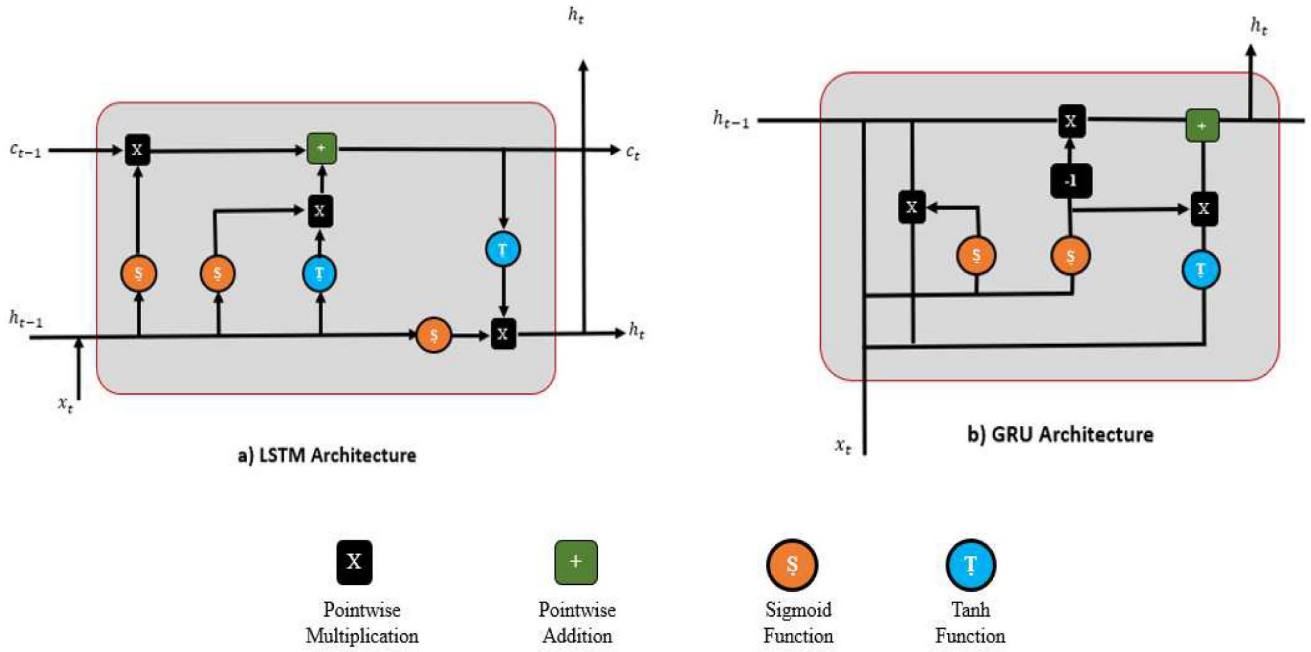

Fig. 2 Exploring architectures as a bidirectional approach for protein sequence identification **a** LSTM configuration and **b** GRU representation

$$r_{t=\sigma}\left(w_{\mathrm{rx}}x_{\mathrm{t}}+u_{\mathrm{rh}}h_{\mathrm{t}-1}\right) \tag{6}$$

$$h_{\mathrm{t}}^{\sim} = \tan\left(w_{\mathrm{hx}}x_{\mathrm{t}} + r_{\mathrm{t}} \odot u_{\mathrm{hh}}h_{\mathrm{t}-1}\right) \tag{7}$$

$$h_{\mathrm{t}} = \left(1 - z_{\mathrm{t}}\right) \odot h_{\mathrm{t}}^{\sim} + z_{\mathrm{t}} \odot h_{\mathrm{t}-1} \tag{8}$$

An improved version of a GRU with a two-layer topology is called a BGRU. Consequently, at any one time, this arrangement gives the output layer access to all contextual data from the input layer [41]. The BGRU's core principle is to process the input sequence both forward and backward, connecting the two outputs in the same output layer [52].

We assess the efficacy of CNN-BiLG for identifying protein sequences. CNN-BiLG draws inspiration from the conventional bidirectional RNN [37], which processes the hidden layer input sequence data both in the forward and backward directions. CNN-BiLG has demonstrated significant outcomes in Speech Recognition [55], Summarization [56], Classification, Energy Consumption Prediction [57], and text generation. The CNN-BiLG structure comprises forward and backward layers as explored by LSTM architectures and GRU representation. This bidirectional approach improves the network's ability to comprehend the context and dependencies in the data by considering both past and future information, as illustrated in Fig. 2.

3 Experiment Setups

We provide a detailed overview of the experimental setup and outcomes, encompassing system configuration, implementation details, evaluation metrics, model training parameters and result comparisons of various models.

3.1 System Configuration and Implementation Details

The models employed for the classification of DBPs are sequential and have been implemented using Python version 3.11.4. The Keras framework version 2.13.1, along with TensorFlow version 2.13.0 as the backend, has been utilized for the implementation. The hardware configuration comprises a Linux Ubuntu 22.04 operating system, an Intel® Core™ i7-9750H CPU @ 2.60 GHz processor, a NVIDIA graphics processing unit (GeForce RTX 2060), and a total of 16.0 GB of RAM. The models are subsequently validated using the hold-out methodology. In which the data are fragmented into training, validation, and testing sets. The training set and validation set are utilized to train the model and validate it during training, whereas the test set is utilized to evaluate the efficacy of the model on data that is yet to be observed. In this paper, we used 80% of the data for training and validation, while the remaining 20% of the data was used for testing.

3.2 Hyperparameter Optimization

We methodically explored several hyperparameter possibilities to maximize the performance of our models, then selected the subset that produced the best results on our dataset. Hyperparameter optimization is the method that allowed us to adapt the models to the specific work of DBP classification. In particular, to maintain alignment and consistency, we zero-filled sequences less than 1000 throughout the coding process to account for the varying lengths of protein sequences in our dataset.

3.3 Model Architecture and Training Parameters

Convolutional neural network (CNN) layers, bidirectional long-short-term memory (LSTM) layers, and fully linked layers were some of the essential parts of our model design. To extract features from the input sequences, we used three 1D CNN layers with different filter and kernel sizes and then clustered the layers. Additionally, temporal dependencies in the data were captured using bidirectional LSTM layers. During model training, we used early stopping, dropping, cross-validation, and self-attention techniques to avoid overfitting. Each model was trained for up to 100 epochs with the Adam optimizer with a batch size of 1024 and a learning rate of 0.001.

3.4 Model Evaluation and Validation

We used a comprehensive set of evaluation criteria, such as sensitivity, specificity, Matthews correlation coefficient (MCC), and overall accuracy, to thoroughly evaluate the performance of our models. Additionally, we have provided a detailed summary of the layers, parameters, and output formats in Table 3 to provide an in-depth understanding of the model architecture. The extensive evaluation and verification process ensured the stability and reliability of our model in accurately categorizing DBPs, which promoted advancements in the fields of molecular biology and bioinformatics.

The detail architecture of the DL model was used to investigate and categorize DNA-binding proteins. It describes the different layers that make up the model: three 1D convolution layers for feature extraction, an embedding layer that transforms input sequences into dense vectors, a spatial dropout layer that specifies the shape and size of the input batch, etc. For increased stability during training, batch normalization layers are included after each convolution layer. Temporal dependencies are captured by bidirectional LSTM layers and relevant input components are highlighted using an attention method. The classification is carried out with dense layers and overfitting is avoided by regulating the dropout. The dimensionality of feature maps is reduced by global average pooling, while binary classification is facilitated by output layers. Based on the conducted experiments, the total params: 590,676 (2.25 MB), the trainable params: 589,780 (2.25 MB) and the non-trainable params: 896 (3.50 KB).

3.5 Evaluation Measures

To evaluate the efficacy of the proposed approach for discerning DBPs solely from primary sequences, multiple assessment measures were employed. The first of these measures was accuracy, a widely employed metric for assessing the performance of classification models. Accuracy measures the ratio of accurately classified instances to the total number of instances in the dataset [1]. In this study, a binary cross-entropy evaluation metric was utilized to calculate the accuracy of the proposed method. Accuracy is measured by the following equation:

$$\mathrm{Ac} = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN})} \tag{9}$$

The second utilized measure is sensitivity, which is the proportion of true positives (correctly identified DBPs) to the total number of actual positives (all DBPs in the dataset). This metric is of particular importance in medical and biological applications, where the cost of false negatives (failure to identify a DNA-binding protein) is significant [58]. It is measured by the following equation:

$$\mathrm{Sensitivity} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \tag{10}$$

The third one, specificity, was employed as an assessment measure. Specificity is the proportion of true negatives (correctly identified non-DBPs) to the total number of actual negatives (all non-DBPs in the dataset). Specificity is a crucial metric in applications where the cost of false positives (identifying a non-DNA-binding protein as a DNA-binding protein) is high [59], which is defined by the following equation:

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{(\mathrm{TN} + \mathrm{FP})} \tag{11}$$

The fourth measure, the Matthews Correlation Coefficient (MCC), is essentially a correlation coefficient between the true and predicted classes and achieves a high value only if the classifier obtains good results in all the entries of the confusion matrix [60]. The MCC is measured by this equation:

$$\mathrm{MCC} = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{12}$$

4 Results and Discussion

The overall results obtained on the dataset are discussed in this section. The present research paper evaluated the efficacy of the proposed method on the dataset taken from [1] and evaluated through the implementation of the hold-out methodology. The results of the conducted experiments demonstrate that the proposed method was able to achieve accuracy with a value of 94%. We have undertaken a thorough comparison of five ML algorithms that are commonly used, namely LR, NB, KNN, DT, and SVM. Based on the conducted experiments, as shown in Table 4, the LR algorithm achieved an accuracy of 0.6813, and the NB algorithm yielded an accuracy of 0.6603. The KNN, DT, and SVM achieved 0.8437, 0.7452, and 0.7832, respectively. Accordingly, the KNN algorithm outperformed all the other ML algorithms with an impressive accuracy of 0.8017. However, the accuracy by KNN is much lower than the high accuracy achieved by the proposed method (0.9401), emphasizing the superiority of our method as compared to the other ML methods.

In Table 4, we explore the performance comparison between our proposed DL model and conventional ML models such as LR, NB, KNN, DT, and SVM. Metrics such as sensitivity, specificity, MCC, and overall accuracy are included in this comprehensive assessment of classification performance and results were verified based on earlier researchers' outcomes [15, 61]. Our proposed model achieves an outstanding sensitivity of 93.88%, which significantly outperforms as compared to LR (67.17%), NB (58.36%), KNN (84.37%), DT (74.52%) and SVM (78.32%). To begin, sensitivity represents the proportion of truly positive predictions among all truly positive cases. This shows how well our DL algorithm can identify good examples of DNA-binding proteins. Then, specificity is measured as the proportion of true negative predictions among all true negative cases. With a specificity of 94.14%, our proposed model outperforms very excellent as compared to LR (69.09%), NB (73.74%), KNN (75.95%), DT (73.56%), and SVM (71.97%) and the results were matched with earlier studies [62, 63]. This shows how reliable our method is in recognizing negative examples, which increases the overall reliability of the classification results.

A good indicator of classification success is the MCC, which considers both true and false positives as well as negatives [64]. With an MCC of 88.02%, our proposed model outperforms as compared to SVM (50.40%), KNN (60.54%), NB (32.48%), LR (36.27%), and DT (48.08%) and results were verified from exploration of earlier researchers [62, 63]. This significant increase in MCC demonstrates how effectively our DL approach balances sensitivity and specificity. Our proposed model achieves an outstanding accuracy of 94.01% when considering the overall accuracy, which is the percentage of correctly identified examples out of the total instances, and results were compared to earlier investigation of Liu [65]. This outperforms LR (68.13%), NB (66.03%), KNN (80.17%), DT (74.04%), and SVM (75.15%), confirming the best performance of the method based on DL and reliability in DNA classification. Binding proteins were analyzed earlier researchers Chen, et al. [66], and we compared our results. The comparative study unequivocally demonstrates the large improvements in DNA-binding protein classification accuracy that our proposed DL model offers compared to conventional ML techniques and results were found excellent in sensitivity, specificity, MCC, and accuracy. Figure 3 shows the confusion matrix for all ML models and the proposed method and results were compared with Nielsen, et al. [67]. Our technique provides a more robust and reliable solution to the difficult question of protein categorization, leading to improvements in molecular biology and bioinformatics. It has greater sensitivity, specificity, MCC, and overall accuracy.

Our experimental investigation involved the implementation of several DL learning models and comparing them with the proposed model and results were matched with earlier purposed methodology [66, 68, 69]. DL learning models which are CNN, LSTM, CNN-LSTM, Deep-CNN and Deep-CNN-LSTM were tested on the DBP dataset and results were compared with earlier researchers [1, 70], and their performance were evaluated based on the four studied measures: sensitivity, specificity, MCC, and accuracy. When compared to the proposed method, it's clear that we achieved the best accuracy (0.9401) using bidirectional LSTM and GRU, in addition to self-attention and results were compared with [71]. The confusion matrices of all the models of DL including our proposed method, are shown in Fig. 4

Based on the above results as mentioned in Table 5, the proposed model was superior to various ML and DL algorithms, emphasizing the effectiveness of our model and results were compared with CNN, LSTM, hybrid CNN-LSTM, Deep-CNN, Deep-CNN-LSTM, CNN, and found excellent metrics results. The efficiency of each model is

Table 4 Comparative analysis and performance evaluation of ML models (LR, NB, KNN, DT, and SVM) vs the proposed model

Metrics/model	LR	NB	KNN	DT	SVM	Proposed
Sensitivity	0.6717	0.5836	0.8437	0.7452	0.7832	0.9388
Specificity	0.6909	0.7374	0.7595	0.7356	0.7197	0.9414
MCC	0.3627	0.3248	0.6054	0.4808	0.5040	0.8802
Accuracy	0.6813	0.6603	0.8017	0.7404	0.7515	0.9401

evaluated using four critical parameters (specificity, sensitivity, MCC, and overall accuracy) that are essential to correctly classify protein-binding proteins [68]. Surprisingly, our proposed model performs best on all metrics, demonstrating its unparalleled power in correctly classifying DNA-binding proteins, and results were compared with earlier researchers [69, 70]. As an example, our model can accurately detect positive cases with a sensitivity of 93.88%, which is much better than the high performance of LSTM (94.03%) and Deep-CNN-LSTM (93 0.67%). Our model achieves a remarkable specificity of 94.14%, which surpasses the scores of all existing deep learning models, such as CNN, LSTM, hybrid CNN-LSTM, and Deep-CNN. With an MCC score of 88.02%, our model outperforms all other DL models and demonstrates our model's superiority in a balanced evaluation of classification performance and results were compared with recent investigations [69, 70]. Moreover, based on the obtained results in [1], which achieved 92.84%, the proposed model enhanced the obtained performance by achieving an accuracy of 94.01%. Moreover, compared with CNN, LSTM, hybrid CNN-LSTM, Deep-CNN, and Deep-CNN-LSTM, our model obtains the highest accuracy of 94.01%, which confirms its robustness and reliability in correctly categorizing DNA-binding proteins. These results demonstrate not only the effectiveness of our proposed DL approach but also its potential to transform molecular biology and bioinformatics by providing a more accurate and reliable solution to the complex problem of protein categorization. The unprecedented performance of our model opens new avenues for medical research and applications by facilitating the knowledge of genetic control processes, drug development, and disease diagnosis.

The comparative analysis presented in Tables 4 and 5 highlights the performance differences between DL models and traditional ML techniques in classifying DNA-binding proteins and results were compared with multiple studies [15, 61, 69, 70]. Although ML techniques such as DT, KNN, LR, NB, and SVM are widely used in bioinformatics, Table 4 shows that their accuracy in classifying DNA binding proteins is limited as compared to our proposed model as per the method of combining techniques. The inability of traditional ML methods to capture the complex patterns observed in protein sequences is demonstrated by the fact that our proposed DL model outperforms KNN, the most accurate ML algorithm. Compared with traditional ML algorithms and other DL architectures such as CNN, LSTM, Deep-CNN, and hybrid CNN-LSTM, Table 5 shows the higher efficiency of deep learning models, including our model. In terms of sensitivity, specificity, MCC, and overall accuracy, our DL model outperforms as compared to others, and results were compared with earlier investigations [15, 61]. Our purpose model explores that DNA-binding proteins

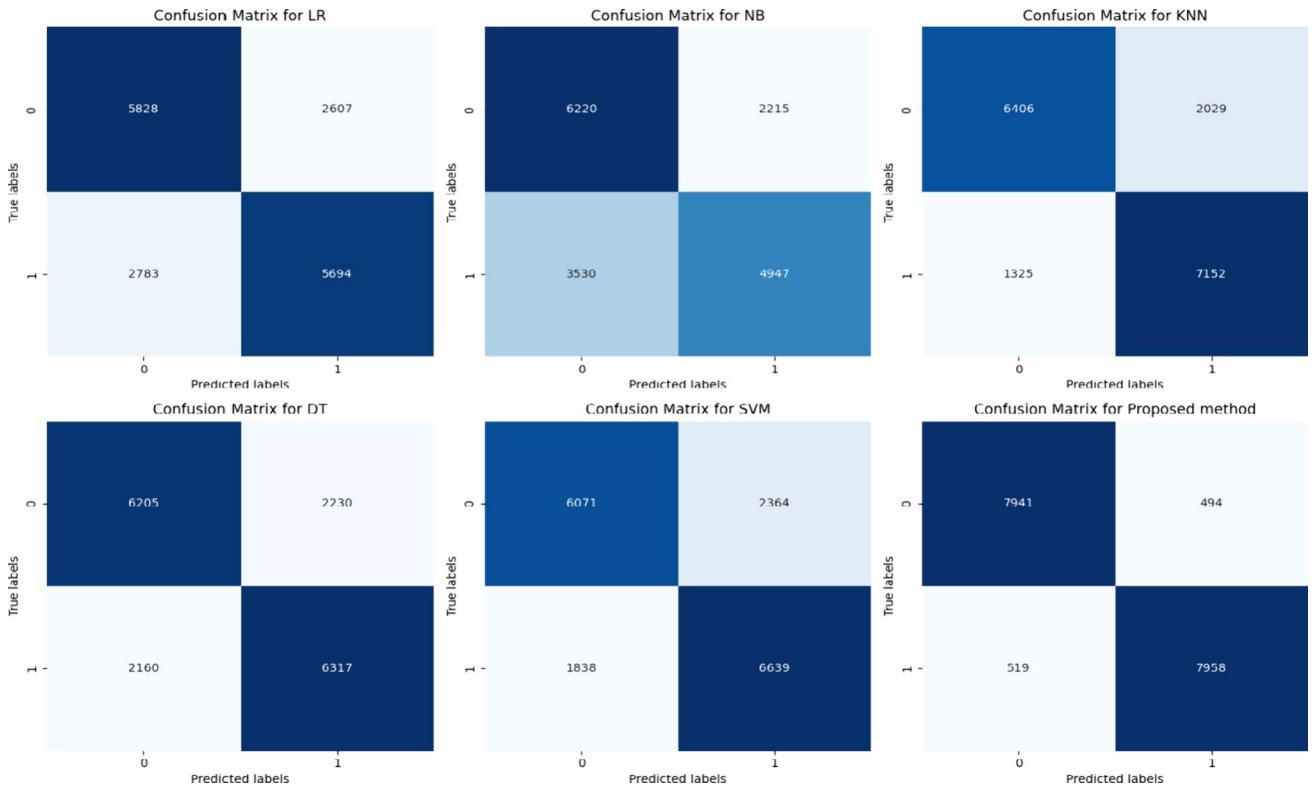

Fig. 3 Comparative confusion matrices of ML models with the proposed model

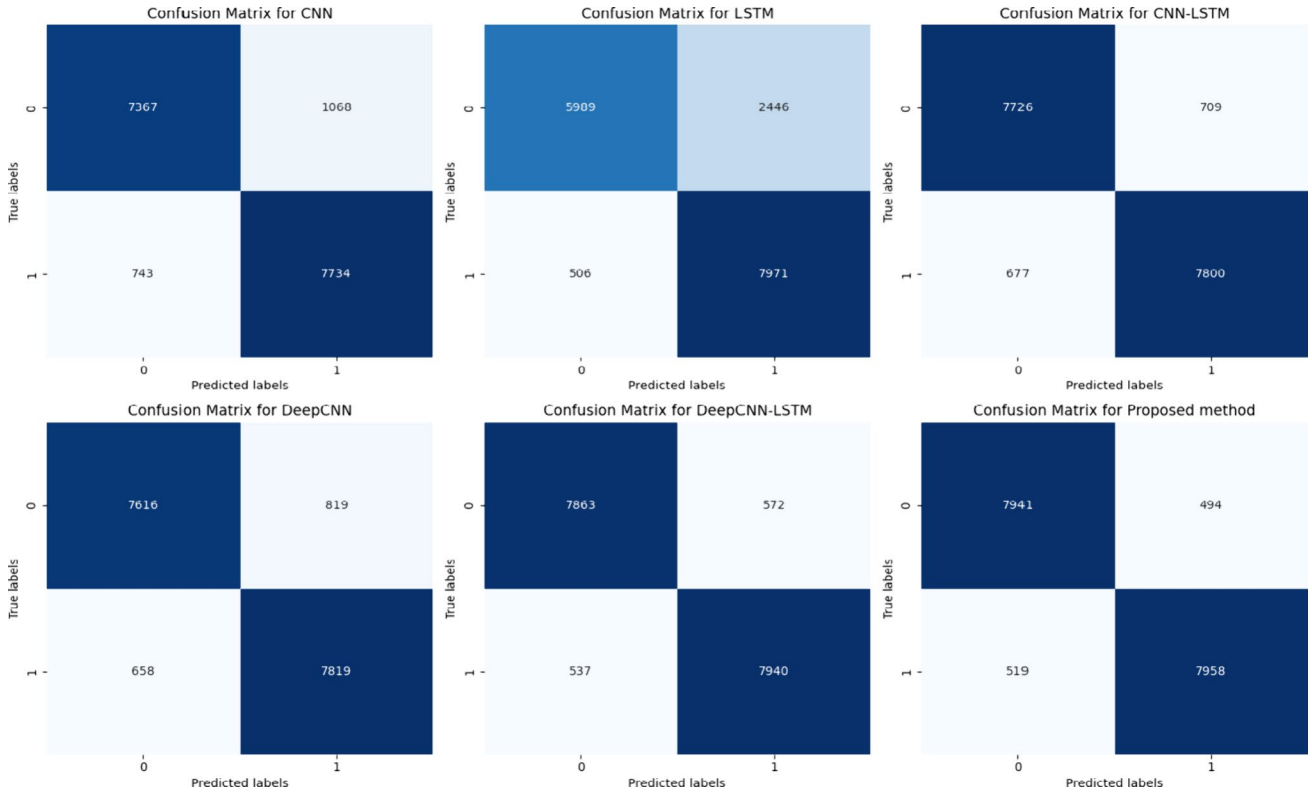

Fig. 4 Comparative confusion matrices of DL models with the proposed model

Table 5 Performance of different DL model's vs the proposed model

Metrics/model	CNN	LSTM	CNN-LSTM	Deep-CNN	Deep-CNN-LSTM	Proposed
Sensitivity	0.9124	0.9403	0.9201	0.9224	0.9367	0.9388
Specificity	0.8734	0.7100	0.9159	0.9029	0.9322	0.9414
MCC	0.7864	0.6686	0.8361	0.8255	0.8689	0.8802
Accuracy	0.8929	0.8254	0.9180	0.9127	0.9344	0.9401

can be consistently and accurately identified. Our study also illustrates how DL techniques can transform molecular biology and bioinformatics by providing more reliable and accurate tools to understand biological processes and improve biomedical research and applications.

5 Conclusion and Future Research Direction

Our research comprehensively evaluates the performance of both traditional ML algorithms and DL models in classifying DNA-binding proteins (DBPs). We utilize a multiple dataset obtained as randomly and employ a hold-out methodology, our proposed DL model achieves an impressive accuracy of 94%, outperforming widely used ML algorithms such as LR, NB, KNN, DT, and SVM. Specifically, our DL model demonstrates superior sensitivity (93.88%), specificity (94.14%), Matthews correlation coefficient (MCC) (88.02%), and overall accuracy (94.01%) compared to these ML algorithms. Furthermore, comparative analysis against various DL models, including CNN, LSTM, CNN-LSTM, Deep-CNN, and Deep-CNN-LSTM, reaffirms the superior performance of our proposed model, highlighting its robustness and reliability in accurately categorizing DBPs. In this paper, we present a novel classification method for the identification of DBPs. We have proposed the CNN-BiLG method, which demonstrates the ability to differentiate proteins rapidly and proficiently, and it autonomously extracts profound characteristics. It enhances the accuracy of predictions and the adaptability of unclassified data. Furthermore, the dataset containing protein sequences has been procured from the Swiss-Prot dataset in the FASTA format and has undergone preprocessing. A variety of ML and DL models have been implemented to evaluate and determine the effectiveness of the proposed model. The conducted comparison indicates that our model

Table 6 Abbreviations used in the paper

DL	Deep learning	BLAST	Basic local alignment search tool
RNN	Recurrent neural network	ML	Machine learning
DBP	DNA binding protein	MCC	Matthews correlation coefficient
NPV	Negative predictive value	RF	Random forest
AI	Artificial intelligence	DT	Decision tree
NB	Naive Bayes	HMM	Hidden Markov model
SVM	Support vector machine	BiLSTM	Bidirectional long short-term Memory
FPR	False positive rate	DNA	Deoxyribonucleic acid
KNN	K-nearest neighbors	PDB	Protein data bank
RNN	Recurrent neural network	CNN	Convolutional neural network

is both effective and efficient, exhibiting commendable classification accuracy. Our proposed model attains 94% accuracy on the dataset. Furthermore, the suggested framework enhances the accuracy of prediction, as well as the fitting of uncharacterized data. The achieved results not only underscore the effectiveness of DL approaches in bioinformatics but also demonstrate the potential of our model to significantly advance molecular biology research and biomedical applications. This study provides insights into the transformative role of DL techniques in understanding biological processes and underscores the importance of further research to explore the integration of additional biological features and advanced techniques like transformer networks to enhance prediction efficacy and broaden the scope of bioinformatics research.

Appendix A

See Table 6 here.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s44196-024-00462-3.

Author Contributions Conceptualization: Nosiba Yousif Ahmed, Wafa Alameen Alsanousi, Eman Mohammed Hamid; data curation: Nosiba Yousif Ahmed, Wafa Alameen Alsanousi, Eman Mohammed Hamid; investigation: Murtada K. Elbashir, Mogtaba Mohammed, Khadija Mohammed Al-Aidarous; methodology: Nosiba Yousif Ahmed, Wafa Alameen Alsanousi, Eman Mohammed Hamid, Khadija Mohammed Al-Aidarous; resources: Murtada K. Elbashir, Mogtaba Mohammed; Software: Nosiba Yousif Ahmed, Khadija Mohammed Al-Aidarous; Visualization: Khadija Mohammed Al-Aidarous; supervision: Murtada K. Elbashir, Mogtaba Mohammed; validation: Nosiba Yousif Ahmed, Khadija Mohammed Al-Aidarous, Eman Mohammed Hamid, Wafa Alameen Alsanousi. All authors have read and agreed to the published version of the manuscript.

Funding The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work, under the General Research Funding program grant code (NU/RG/SERC/12/28).

Data Availability All relevant data are within the paper and its Supplementary information files.

Declarations

Conflict of interest The authors declare no conflict of interest.

Institutional Review Board Statement Not applicable.

References

1. Qu, Y.-H., Yu, H., Gong, X.-J., Xu, J.-H., Lee, H.-S.: On the prediction of DNA-binding proteins only from primary sequences: a deep learning approach. PLoS ONE **12**, e0188129 (2017)
2. Raghava, G.P, Gromiha, M.M., Kumar, M.: Identification of DNA-binding proteins using support vector machines and evolutionary profiles. BMC Bioinform.**8**, 463 (2007)
3. Zhao, Z., Yang, W., Zhai, Y., Liang, Y., Zhao, Y.: Identify DNA-binding proteins through the extreme gradient boosting algorithm. Front. Genet. **12**, 821996 (2022)
4. Li, H., Long, C., Xiang, J., Liang, P., Li, X., Zuo, Y.: Dppa2/4 as a trigger of signaling pathways to promote zygote genome activation by binding to CG-rich region. Brief. Bioinform. **22**, bbaa342 (2021)
5. Barukab, O., Ali, F., Khan, S.A.: DBP-GAPred: an intelligent method for prediction of DNA-binding proteins types by enhanced evolutionary profile features with ensemble learning. J. Bioinform. Comput. Biol. **19**, 2150018 (2021)
6. Luscombe, N.M., Austin, S.E., Berman, H.M., Thornton, J.M.: An overview of the structures of protein-DNA complexes. Genome Biol. **1**, 1–37 (2000)

7. Stawiski, E.W., Gregoret, L.M., Mandel-Gutfreund, Y.: Annotating nucleic acid-binding function based on protein structure. J. Mol. Biol. **326**, 1065–1079 (2003)
8. Mishra, A., Pokhrel, P., Hoque, M.T.: StackDPPred: a stacking based prediction of DNA-binding protein from sequence. Bioinformatics **35**, 433–441 (2019)
9. Guo, J.-T., Malik F. J. B.: Single-Stranded DNA binding proteins and their identification using machine learning-based approaches. Biomol. **12**, 1187 (2022)
10. Zafar, I., Anwar, S., Yousaf, W., Nisa, F.U., Kausar, T., Ul Ain, Q., et al.: Reviewing methods of deep learning for intelligent healthcare systems in genomics and biomedicine. Biomed. Signal Process. Control **86**, 105263 (2023)
11. Chen, J., Gu, Z., Lai, L., Pei, J.: In silico protein function prediction: the rise of machine learning-based approaches. Med. Rev. **3**, 487–510 (2023)
12. Narykov, O.: Modern computer science approaches in biology: from predicting molecular functions to modeling protein structure. University of Virginia, (2022)
13. Zeng, Y., Gong, M., Lin, M., Gao, D., Zhang, Y.J.I.A.: A review about transcription factor binding sites prediction based on deep learning. IEEE Access **8**, 219256–219274 (2020)
14. Koo, P.K., Ploenzke, M.: Deep learning for inferring transcription factor binding sites. Curr. Opin. Syst. Biol. **19**, 16–23 (2020)
15. Lou, W., Wang, X., Chen, F., Chen, Y., Jiang, B., Zhang, H.: Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naive Bayes. PLoS ONE **9**, e86703 (2014)
16. Brown, J., Akutsu, T.: Identification of novel DNA repair proteins via primary sequence, secondary structure, and homology. BMC Bioinform. **10**, 1–22 (2009)
17. Dobbs, D., Yan, C., Terribilini, M., Wu, F., Jernigan, R., Honavar, V.: Predicting DNA-binding sites of proteins from amino acid sequence. Int J Mol Sci. (2006). https://doi.org/10.3390/ijms16035194
18. Zhu, H.: Big data and artificial intelligence modeling for drug discovery. Annu. Rev. Pharmacol. Toxicol. **60**, 573–589 (2020)
19. Liu, J., Li, J., Wang, H., Yan, J.J.S.C.L.S.: Application of deep learning in genomics. Sci. China Life Sci. **63**, 1860–1878 (2020)
20. Zeng, H., Edwards, M.D., Liu, G., Gifford, D.K.: Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics **32**, i121–i127 (2016)
21. Glasscock, C.J, Pecoraro, R., McHugh, R., Doyle, L.A., Chen, W., Boivin, O. et al.: Computational design of sequence-specific DNA-binding proteins. bioRxiv. (2023)
22. Li, G., Du, X., Li, X., Zou, L., Zhang, G., Wu, Z.: Prediction of DNA binding proteins using local features and long-term dependencies with primary sequences based on deep learning. PeerJ **9**, e11262 (2021)
23. Zhou, C., Yu, H., Ding, Y., Guo, F., Gong, X.-J.: Multi-scale encoding of amino acid sequences for predicting protein interactions using gradient boosting decision tree. PLoS ONE **12**, e0181426 (2017)
24. Jia, Y., Huang, S., Zhang, T.: KK-DBP: a multi-feature fusion method for DNA-binding protein identification based on random forest. Front. Genet. **12**, 811158 (2021)
25. Qian, Y., Jiang, L., Ding, Y., Tang, J., Guo, F.: A sequence-based multiple kernel model for identifying DNA-binding proteins. BMC Bioinform. **22**, 1–18 (2021)
26. Sang, X., Xiao, W., Zheng, H., Yang, Y., Liu, T.: HMMPred: accurate prediction of DNA-binding proteins based on HMM profiles and XGBoost feature selection. Comput. Math. Methods Med. (2020). https://doi.org/10.1155/2020/1384749
27. Wang, J., Zheng, H., Yang, Y., Xiao, W., Liu, T.: PredDBP-stack: prediction of DNA-binding proteins from HMM profiles using a stacked ensemble method. BioMed Res. Int. (2020). https://doi.org/10.1155/2020/7297631
28. Ma, X., Guo, J., Sun, X.: DNABP: identification of DNA-binding proteins based on feature selection using a random forest and predicting binding residues. PLoS ONE **11**, e0167345 (2016)
29. Zou, Y., Wu, H., Guo, X., Peng, L., Ding, Y., Tang, J., et al.: MK-FSVM-SVDD: a multiple kernel-based fuzzy SVM model for predicting DNA-binding proteins via support vector data description. Curr. Bioinform. **16**, 274–283 (2021)
30. Ali, F., Kabir, M., Arif, M., Swati, Z.N.K., Khan, Z.U., Ullah, M., et al.: DBPPred-PDSD: machine learning approach for prediction of DNA-binding proteins using discrete wavelet transform and optimized integrated features space. Chemom. Intell. Lab. Syst. **182**, 21–30 (2018)
31. Liu, X.-J., Gong, X.-J., Yu, H., Xu, J.-H.: A model stacking framework for identifying DNA binding proteins by orchestrating multi-view features and classifiers. Genes **9**, 394 (2018)
32. Wei, L., Tang, J., Zou, Q.: Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information. Inf. Sci. **384**, 135–144 (2017)
33. Zaman, R., Chowdhury, S.Y., Rashid, M.A., Sharma, A., Dehzangi, A., Shatabda, S.: HMMBinder: DNA-binding protein prediction using HMM profile based features. BioMed Res. Int. (2017). https://doi.org/10.1155/2017/4590609
34. Xu, R., Zhou, J., Wang, H., He, Y., Wang, X., Liu, B.: Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. BMC Syst. Biol. (2015). https://doi.org/10.1186/1752-0509-9-S1-S10
35. Öncül, A.B.: LSTM-GRU based deep learning model with Word-2Vec for transcription factors in primates. Balkan J. Electr. Comput. Eng. **11**, 42–49 (2023)
36. Tayara, H., Chong, K.T.: Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network. Sensors **18**, 3341 (2018)
37. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**, 2673–2681 (1997)
38. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
39. Liu, X.: Deep recurrent neural network for protein function prediction from sequence arXiv preprint arXiv:1701.08318 (2017) 28 Jan
40. Lu, W., Zhou, N., Ding, Y., Wu, H., Zhang, Y., Fu, Q., et al.: Application of DNA-binding protein prediction based on graph convolutional network and contact map. BioMed Res. Int. (2022). https://doi.org/10.1155/2022/9044793
41. Priyadarshini, I., Cotton, C.: A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis. J. Supercomput. **77**, 13911–13932 (2021)
42. Du, X., Diao, Y., Liu, H., Li, S.: MsDBP: exploring DNA-binding proteins by integrating multiscale sequence information via Chou's five-step rule. J. Proteome Res. **18**, 3119–3132 (2019)
43. Hu, S., Ma, R., Wang, H.: An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences. PLoS ONE **14**, e0225317 (2019)
44. Khan, S.U., Baik, R.: MPPIF-net: identification of plasmodium falciparum parasite mitochondrial proteins using deep features with multilayer Bi-directional LSTM. Processes **8**, 725 (2020)
45. Xie, J., Zheng, J., Hong, X., Tong, X., Liu, X., Song, Q., et al.: Protein-DNA complex structure modeling based on structural template. Biochem. Biophys. Res. Commun. **577**, 152–157 (2021)
46. Yan, J., Jiang, T., Liu, J., Lu, Y., Guan, S., Li, H., et al.: DNA-binding protein prediction based on deep transfer learning. Math. Biosci. Eng. **19**, 7719–7736 (2022)
47. Yadav, M., Yadav, H.S.: Biochemistry: Fundamentals and Bioenergetics. Bentham Science Publishers, Sharjah (2021)

48. Song, L., Li, D., Zeng, X., Wu, Y., Guo, L., Zou, Q.: nDNA-prot: identification of DNA-binding proteins based on unbalanced classification. BMC Bioinform. **15**, 1–10 (2014)
49. Pan, G., Wang, J., Zhao, L., Hoskins, W., Tang, J.: Computational methods for predicting DNA binding proteins. Curr. Proteom. **17**, 258–270 (2020)
50. Trabelsi, A., Chaabane, M., Ben-Hur, A.: Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. Bioinformatics **35**, i269–i277 (2019)
51. Bairoch, A., Apweiler, R.: The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. Nucleic Acids Res. **27**, 49–54 (1999)
52. Alsanousi, W.A., Ahmed, N.Y., Hamid, E.M., Elbashir, M.K., Musa, M.E.M., Wang, J., et al.: A novel deep learning-assisted hybrid network for plasmodium falciparum parasite mitochondrial proteins classification. PLoS ONE **17**, e0275195 (2022)
53. Wang, L., Wang, H.-F., Liu, S.-R., Yan, X., Song, K.-J.: Predicting protein-protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. Sci. Rep. **9**, 9848 (2019)
54. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C. et al.: CNN architectures for large-scale audio classification, in 2017 IEEE international conference on acoustics, speech and signal processing (icassp), 2017, pp. 131–135.
55. Mustaqeem, Kwon, S.: A CNN-assisted enhanced audio signal processing for speech emotion recognition. Sensors **20**, 183 (2019)
56. Hussain, T., Muhammad, K., Ullah, A., Cao, Z., Baik, S.W., de Albuquerque, V.H.C.: Cloud-assisted multiview video summarization using CNN and bidirectional LSTM. IEEE Trans. Ind. Inf. **16**, 77–86 (2019)
57. Ullah, F.U.M., Ullah, A., Haq, I.U., Rho, S., Baik, S.W.: Short-term prediction of residential power energy consumption via CNN and multi-layer bi-directional LSTM networks. IEEE Access **8**, 123369–123380 (2019)
58. Vujović, Ž: Classification model evaluation metrics. Int. J. Adv. Comput. Sci. Appl. **12**, 599–606 (2021)
59. Monaghan, T.F., Rahman, S.N., Agudelo, C.W., Wein, A.J., Lazar, J.M., Everaert, K., et al.: Foundational statistical principles in medical research: sensitivity, specificity, positive predictive value, and negative predictive value. Medicina **57**, 503 (2021)
60. Hicks, S.A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M.A., Halvorsen, P., et al.: On evaluation metrics for medical applications of artificial intelligence. Sci. Rep. **12**, 5979 (2022)
61. Yan, C., Terribilini, M., Wu, F., Jernigan, R.L., Dobbs, D., Honavar, V.: Predicting DNA-binding sites of proteins from amino acid sequence. BMC Bioinform. **7**, 1–10 (2006)
62. Al-Ajlan, A., El Allali, A.: Feature selection for gene prediction in metagenomic fragments. BioData Min. **11**, 1–12 (2018)
63. Shoombuatong, W., Mekha, P., Chaijaruwanich, J.J.: Sequence based human leukocyte antigen gene prediction using informative physicochemical properties. Int. J. Data Min. Bioinform. **13**, 211–224 (2015)
64. Cao, C., Chicco, D., Hoffman, M.M.: The MCC-F1 curve: a performance evaluation technique for binary classification. arXiv preprint arXiv:2006.11278 (2020)
65. Liu, B.: BioSeq-analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. Brief. Bioinform. **20**, 1280–1294 (2019)
66. Chen, Z., Zhao, P., Li, F., Marquez-Lago, T.T., Leier, A., Revote, J., et al.: iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. Brief. Bioinform. **21**, 1047–1057 (2020)
67. Nielsen, H., Brunak, S., von Heijne, G.: Machine learning approaches for the prediction of signal peptides and other protein sorting signals. Protein Eng. Des. Sel. **12**, 3–9 (1999)
68. Qu, K., Wei, L., Zou, Q.: A review of DNA-binding proteins prediction methods. Curr. Bioinform. **14**, 246–254 (2019)
69. Rube, H.T., Rastogi, C., Feng, S., Kribelbauer, J.F., Li, A., Becerra, B., et al.: Prediction of protein–ligand binding affinity from sequencing data with interpretable machine learning. Nat. Biotechnol. **40**, 1520–1527 (2022)
70. Das, S., Chakrabarti, S.J.: Classification and prediction of protein–protein interaction interface using machine learning algorithm. Sci. Rep. **11**, 1761 (2021)