A Machine Learning Method for Prediction of DNA Binding Proteins

Sufal Das
Assistant Professor
Department of Information Technology
North-Eastern Hill University
Shillong Meghalaya India -793022

Email: sufal.das@gmail.com
ORCID ID:- https://orcid.org/ 0000-0002-0998-2050

Abstract— DNA binding proteins attach to DNA. Most of the important biological functions including modulating protein creation, controlling cell division and cell growth, and transcription are performed by these DNA binding proteins. So, it is very important to know what type of protein it is and what instructions it contains for building and maintaining the human cells. Therefore, using DNA binding proteins, diseases like cancers and genetic diseases can be cured by developing some essential drugs against these diseases. Some examples of DNA binding proteins are transcription factors that bind to particular area in the DNA for modulating transcription. Transcription is the operation of duplicating DNA to RNA. Identification of DNA binding protein by efficient methods with great performance is a very challenging task for researchers. Very costly and more time consuming experimental methods including X-ray crystallography[14], filter binding assays [18], chromatin immuno precipitation on microarrays [12], genetic analysis [19] etc. exist. But, nowadays researchers apply computational methods that are less time consuming and also inexpensive for the prediction of DNA binding protein [10].

Keywords: Protein, DNA-binding Protein, Machine Learning, Prediction, Deep Learning.

DOI: 10.48047/ecb/2023.12.si4.973

I. INTRODUCTION

DNA binding proteins attach to DNA. Most of the important biological functions including modulating protein creation, controlling cell division and cell growth, and transcription are performed by these DNA binding proteins. So, it is very important to know what type of protein it is and what instructions it contains for building and maintaining the human cells. Therefore, using DNA binding proteins, diseases like cancers and genetic diseases can be cured by developing some essential drugs against these diseases. Some examples of DNA binding proteins are transcription factors that bind to particular area in the DNA for modulating transcription. Transcription is the operation of duplicating DNA to RNA. Identification of DNA binding protein by efficient methods with great performance is a very challenging task for researchers. Very costly and more time consuming experimental methods including X-ray crystallography[14], filter binding assays [18], chromatin immuno precipitation on microarrays [12], genetic analysis [19] etc. exist. But, nowadays researchers apply computational methods that are less time consuming and also inexpensive for the prediction of DNA binding protein [10].

II. RELATED WORKS

A. PseAAC, a model for identifying DNA-binding protein [3].

To identify DNA binding protein M.Saifur Rahman et al.[3] first took out all the significant information from the protein strings. After getting the features, they ranked the features using Random Forest model and Recursive Feature Elimination[5] method. Finally, all the features were given to the support vector machine[7] as an input for prediction.

PDB1075 and PDB186 datasets were used where PDB1075 includes 525 positive samples and 550 negative samples and PDB186 includes 93 equal number of positive and negative samples. This method acquired an accuracy of 95.91%, specificity of 97.64%, sensitivity of 94.10% and MCC of 0.92 on PDB1075 dataset. The method acquired an accuracy of 77.42%, specificity of 70.97%, sensitivity of 83.87% and MCC of 0.553 on PDB186 dataset.

The method shows good performance. This method generates more feature dimensions, hence it is complex. Ensemble classifier needs more time and space.

B. DNA-binding protein prediction found on PSSM information [2].

Three methods NMBAC, PSSM-DWT and PSSM-DCT were used by Yubo Wang et al.[2] to take out the features from the protein string. And these selected features were given to SVM as input for training the classifier.

PDB1075, PDB594 and PDB186 datasets were used where PDB1075 includes 525 positive samples and 550 negative samples, PDB594 includes 297 equal number of positive and negative samples. Dataset PDB186 also includes 93 equal number of positive and negative samples. This method acquired an accuracy of 86.23%, specificity of 85.09%, sensitivity of 87.43% and MCC of 0.73 on PDB1075 dataset. The method acquired an accuracy of 76.3%, specificity of 60.2%, sensitivity of 92.5% and MCC of 0.557 on PDB186 dataset.

Merged feature extraction techniques give better performance. Performance of PSSM-DCT features are poor.

C. DNA-Binding Protein identifying using Evolutionary [1].

For identifying DNA-Binding Protein Xiangzheng FU et al.[1] have used a feature extraction method named K-PSSM Composition. The evolutionary detail of a protein is conserved by those extracted features during the process of evolution. To extract the optimal set of features RFE methods are used and it is used for training the support vector machine.

PDB1075 and PDB186 were used where PDB1075 includes 525 positive samples and 550 negative samples. Dataset PDB186 also includes 93 equal numbers of positive and negative samples. This method acquired an accuracy of 89.77%, specificity of 89.27%, sensitivity of 90.29% and MCC of 0.80 on PDB1075 dataset. The method acquired an accuracy of 88.71%, specificity of 81.72%, sensitivity of 95.70% and MCC of 0.78 on PDB186 dataset.

Time and space taken is less. Production is not so good. Feature selection is required for good performance.

D. A Model stack Framework to identify DNA Binding Protein [4]

For identifying DNA Binding Protein Xiu-Juan Liu et al.[4] have put forward a model stack framework to integrate and evaluate freely-coupled models by MSFBinder. Feature extraction techniques such as

ACStruc, 188D, PSSM DWT, and Local DPP are combined in this framework which are used to train random forest and SVM. After that for the final prediction a logistic regression model was enforced.

PDB1075 and PDB186 were used where PDB1075 includes 525 positive samples and 550 negative samples. Dataset PDB186 also includes 93 equal number of positive and negative samples. This method acquired an accuracy of 83.53%, specificity of 83.27%, sensitivity of 83.81% and MCC of 0.67 on PDB1075 dataset. The method acquired an accuracy of 81.72%, specificity of 74.19%, sensitivity of 89.25% and MCC of 0.64 on PDB186 dataset.

Better performance is acquired with Local DPP. The execution of AC struct is superior than the others.

E. DNA-binding proteins prediction using revolutionary profiles and SVM [18].

For identifying DNA Binding Protein Manish Kumar et al.[18] have used PSSM profiles by working with SVM.

DNAaset is used which consists of 1153 positive and negative samples. Using dipeptide compositions accuracy of 71.59% with Sensitivity = 72.59% , Specificity = 70.59%, MCC = 0.43 are obtained and using amino acids accuracy of 72.42% with Sensitivity = 72.51% , Specificity = 72.33%, MCC = 0.45 are obtained.PSSM profiles increased the performance with an Accuracy of 74.22% with Sensitivity = 73.53% , Specificity = 74.92%, MCC = 0.49. Again, on DNAset an SVM model has been developed, which comprises 146 numbers of DNA- binding and 250 numbers of non-DNA binding and it attained an Accuracy of 86.62% with sensitivity = 86.32% , Specificity = 86.80%, MCC = 0.72using PSSM profiles and by using amino acid composition accuracy of 79.80%, Sensitivity = 78.11% , Specificity = 80.80%, MCC = 0.58 is obtained.

III. PROPOSED METHODOLOGY

This section describes the detailed methodology for the prediction of DNA-binding protein using Convolutional Neural Network(CNN) [9, 11]. The steps dataset description, classification and performance evaluation are described as below :

In the proposed method, a convolution filter is used to produce a feature map. A convolutional filter or kernel size of 3*3 is used on this convolution layer. Strides = (1, 1) is used which defines how much the convolution filter is shifted at each step and padding = ‘valid’ is used here which represents zero padding or no padding. The CNN [15] architecture of the proposed work is illustrated by Fig. 1 where I have used three convolution layers such as C1, C2, C3 with a filter size of 3*3. Again a pooling window of 2*2 is used with two subsampling layers S1, S2 .

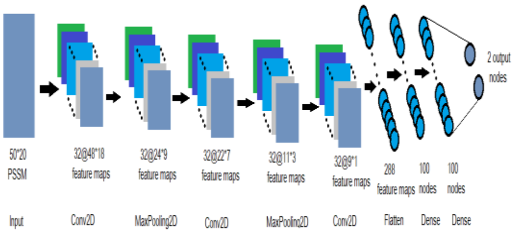

Fig.1: CNN architecture of the proposed model

In first Convolution Layer (C1) ,zero padding and stride =1 is used to the input matrix I with size 50*20 convolved with 32 kernels which generated (50-3 +1) * (20-3 +1) = 48*18 sized 32 convolved features. Hence, I get (3*3+1) * 32 = 320 total parameters for this functioning. Learning of these parameters will be in the training phase. To maintain non-linearity in the matrix, a activation function used in every convolution layer is ReLU (Rectified inear Unit). The generated features by the C1 layer are provided into first subsampling layer (S1). Max pooling is used with a window size of 2 * 2 in this subsampling layer. Hence, 32 pooled feature maps each of size 24*9 are generated.

In second Convolution Layer (C2), I use 32 kernels on 32 number of features map that are generated by previous layer and it produces same number of convolved features map each of size (24-3 +1) * (9-3 +1) = 22*7. Hence, I get (32*3*3 +1) * 32 = 9248 total parameters for this functioning. The generated features by the C2 layer are provided into second subsampling layer (S2). Max pooling is used with a window size of 2 * 2 in this sub-sampling layer. Hence, 32 pooled feature maps each of size 11*3 are generated.

In third Convolution Layer (C3), I use 32 kernels on 32 number of features map that are generated by previous layer and it produces same number of convolved features map each of size (11-3 +1) * (3-3 +1) = 9*1. Hence, I get (32*3*3 +1) * 32 = 9248 total parameters for this functioning.

Thereafter a Flatten Layer is used to flatten the output that produced a vector size of (9*1*32)=288.

Then I provide the produced features map to a Dense Layer with 100 nodes and "ReLu" activation. Here, I get (288*100)+100 = 28900 total parameters for this functioning.

To minimize the overfitting during training phase, I add a dropout layer with a dropout rate of 20% . An another Dense Layer is used with same number of nodes and same activation that results (100*100)+100 = 10100 number of total parameters. Again one more dropout layer with same dropout rate is used.

Finally, one more Dense Layer with 2 nodes and "Sigmoid" activation is used for the classification. Hence, I get (100*2) + 2 = 202 number of total parameters for this functioning. Table 1 shows the number of trainable parameters in each layer and the total parameters for the model.

Layers	Parameters
Conv2D(C1)	320
Conv2D(C2)	9,248
Conv2D(C3)	9,248
Dense(1)	28,900
Dense(2)	10,100
Dense(3)	202
Total	**58,018**

Table 1 : Trainable Parameters for Proposed Model

The performance parameters namely accuracy, precision, sensitivity or recall, specificity, F1-Score and the confusion matrix are used to evaluate the performance of the proposed CNN [16] model. The parameters are determined by calculating the number of True Positives (TP), False Positives(FP),True Negatives(TN) and False Negatives(FN).

- True Positive (TP) : DNA-binding Proteins correctly classified as DNA-binding Proteins.
- False Negative (FN) : DNA-binding Proteins incorrectly classified as non DNA-binding Proteins.
- True Negative (TN) : Non DNA-binding Proteins correctly classified as non DNA-binding Proteins.
- False Positive (FP) : Non DNA-binding Proteins incorrectly classified as DNA-binding Proteins.

The following formulas are used for determining Accuracy, Precision, Sensitivity or Recall, Specificity, and F1-Score [6,8].

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \qquad \text{(Eq. No. 1)}$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \qquad \text{(Eq. No. 2)}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \times 100\% \qquad \text{(Eq. No. 3)}$$

$$\text{Specificity} = \frac{TN}{TN+FP} \times 100\% \qquad \text{(Eq. No. 4)}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Sensitivity}}{\text{Precision} \times \text{Sensitivity}} \times 100\% \qquad \text{(Eq. No. 5)}$$

IV. RESULT ANALYSIS

After building the proposed CNN model, 'Adam' optimizer with 'categorical crossentropy' loss function have been used for model compilation. A batch size of 62 and 3 epochs are used to fit the training data of PDB186 and a batch size of 215 and 5 epochs are used to fit the training data of PDB1075 with the proposed model.

The proposed method has acquired a training accuracy of 90.77% with a loss of 33.53% and a validation accuracy of 91.07% with a loss of 24.83% when performed on the PDB186 dataset, and it acquired a training accuracy of 94.41% with a loss of 19.00% and a validation accuracy of 97.83 % with a loss of 13.67% on the PDB1075 dataset. Fig.2 shows the graphs for model with accuracy and model loss against different epochs on different datasets.

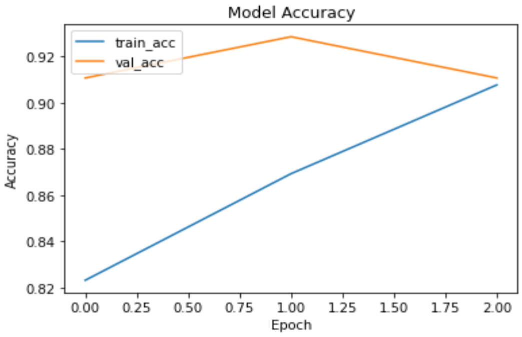

Fig. 1. Model Accuracy on PDB186 dataset

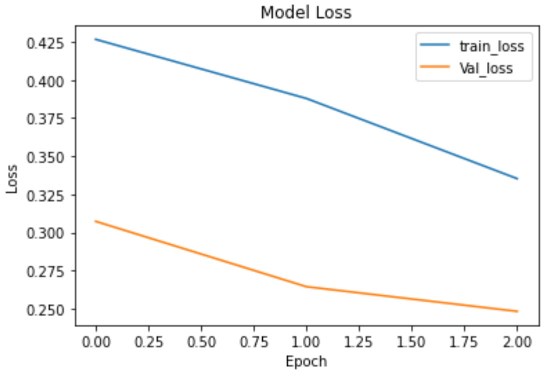

Fig. 2. Model Loss on PDB186 dataset

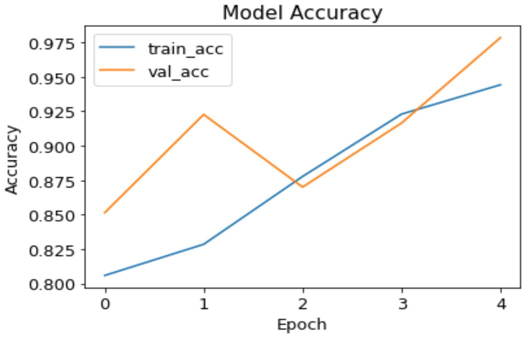

Fig. 3. Model Accuracy on PDB1075 dataset

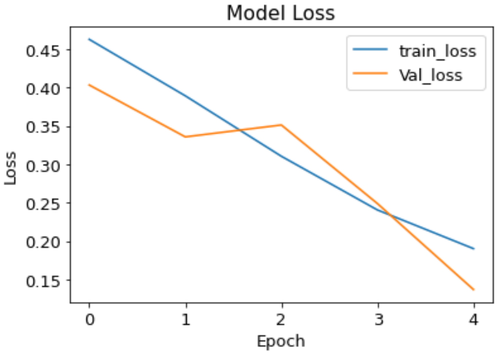

Fig. 4. Model Loss on PDB1075 dataset

The confusion matrix[17] for the proposed CNN model on PDB186 has been represented by the table 1. From PDB186 dataset 56 numbers of DNA-binding and non DNA-binding proteins have been used to validate the model. From 28 non DNA-binding proteins, the model can accurately classify 25 as non DNA-binding proteins while 3 as DNA-binding proteins. And from another 28 DNA-binding proteins, it can accurately classify 26 as DNA-binding proteins while 2 as non DNA-binding proteins. An another confusion matrix[13] for the model on PDB1075 has been represented by the table 2. From PDB1075 dataset 323 numbers of DNA-binding and non DNA-binding proteins have been used to validate the model. From 151 non DNA-binding proteins, the model can accurately classify 151 as non DNA-binding proteins while 0 as DNA-binding proteins. And from 172 DNA-binding proteins, it can accurately classify 165 as DNA-binding proteins while 7 as non DNA-binding proteins.

	Predicted Class 1 (non-dbp)	**Predicted Class 2 (dbp)**
Predicted Class 1 (non-dbp)	25	3
Predicted Class 2 (dbp)	2	26

Table 2: Confusion Matrix on PDB186

	Predicted Class 1 (non-dbp)	**Predicted Class 2 (dbp)**
Predicted Class 1 (non-dbp)	151	21
Predicted Class 2 (dbp)	7	165

Table 3: Confusion Matrix on PDB1075

Table 4 shows the performance parameters for the proposed model for evaluation. It illustrates that accuracy, precision, recall or sensitivity, specificity and f1-score are used to evaluate the model.

Dataset	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Specificity (%)**	**F1-Score (%)**
PDB186	91.07	89.66	92.86	89.29	91.23
PDB1075	97.83	100	95.93	100	97.92

Table 4: Performance Parameters for the Model

Table 5 and 6 illustrate the comparison between several methods. It has shown that the proposed model has given the highest accuracy (91.07%), and specificity (89.29%) values on PDB186 and specificity(100%)

values on PDB1075 among the different methods.

For better understanding, two bar graphs (Fig. 5 and Fig. 6) have also been plotted to compare only the accuracy values on different datasets of the proposed method with the existing methods.

Method	**Accuracy (%)**	**Recall (%)**	**Specificity (%)**
Xiu-Juan Liu et al. [4]	81.72	89.25	74.19
Yubo Wang et al. [2]	76.3	92.5	60.2
X.Fu et al. [1]	88.71	95.7	81.72
DPP-PseAAC [3]	77.42	83.87	70.97
Proposed Method	91.7	92.86	89.29

Table 5 : Comparison with Existing Methods on PDB186

Method	**Accuracy (%)**	**Recall (%)**	**Specificity (%)**
Xiu-Juan Liu et al. [4]	88.84	85.52	84.18
Yubo Wang et al. [2]	86.23	87.43	85.09
X.Fu et al. [1]	89.77	90.29	89.27
DPP-PseAAC [3]	95.91	94.1	97.04
Proposed Method	97.83	95.93	98.01

Table 6 : Comparison with Existing Methods on PDB1075

V. CONCLUSION

In this paper various existing related works are demonstrated as well as analyzed with advantages and limitations. Finally, an efficient prediction method to identify DNA-binding protein using Convolutional Neural Networks (CNN) is proposed. The execution of the prediction of DNA binding protein relies on how we take out the evolutionary detail of these proteins. Here, various experiments are demonstrated and analyzed. Also, we have seen the advantages and disadvantages of related works. Finally, I have introduced a new technique to predict DNA-binding protein using Convolutional Neural Networks. And we have seen that it gives a better performance than the other methods.

REFERENCES

[1] Fu, Xiangzheng, Wen Zhu, Bo Liao, Lijun Cai, Lihong Peng, and Jialiang Yang. "Improved DNA-binding protein identification by incorporating evolutionary information into the Chou’s PseAAC." *IEEE Access* 6 (2018): 66545-66556.

[2] Wang, Yubo, Yijie Ding, Fei Guo, Leyi Wei, and Jijun Tang. "Improved detection of DNA-binding proteins via compression technology on PSSM information." PloS one 12, no. 9 (2017): e0185587.

[3] Rahman, M. Saifur, Swakkhar Shatabda, Sanjay Saha, Mohammad Kaykobad, and M. Sohel Rahman. "DPP-PseAAC: a DNA-binding protein prediction model using Chou’s general PseAAC." Journal of theoretical biology 452 (2018): 22-34.

[4] Liu, Xiu-Juan, Xiu-Jun Gong, Hua Yu, and Jia-Hui Xu. "A model stacking framework for identifying DNA binding proteins by orchestrating multi-view features and classifiers." Genes 9, no. 8 (2018): 394.

[5] Qiu, Wang-Ren, Bi-Qian Sun, Xuan Xiao, Zhao-Chun Xu, and Kuo-Chen Chou. "iHyd-PseCp: Identify hydroxyproline and hydroxylysine in proteins by incorporating sequence-coupled effects into general PseAAC." Oncotarget 7, no. 28 (2016): 44310.

[6] Yu, Bin, Lifeng Lou, Shan Li, Yusen Zhang, Wenying Qiu, Xue Wu, Minghui Wang, and Baoguang Tian. "Prediction of protein structural class for low-similarity sequences using Chou’s pseudo amino acid composition and wavelet denoising." Journal of Molecular Graphics and Modelling 76 (2017): 260-273.

[7] Zou, Chuanxin, Jiayu Gong, and Honglin Li. "An improved sequence based prediction protocol for DNA-binding proteins using SVM and comprehensive feature analysis." BMC bioinformatics 14, no. 1 (2013): 1-14.

[8] Liu, Bin, Junjie Chen, and Xiaolong Wang. "Protein remote homology detection by combining Chou’s distance-pair pseudo amino acid composition and principal component analysis." Molecular Genetics and Genomics 290, no. 5 (2015): 1919-1931.

[9] Pan, Xiaoyong, Peter Rijnbeek, Junchi Yan, and Hong-Bin Shen. "Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks." BMC genomics 19, no. 1 (2018): 1-11.

[10] Zhao, Huiying, Yuedong Yang, and Yaoqi Zhou. "Structure-based prediction of RNA-binding domains and RNA-binding sites and application to structural genomics targets." Nucleic acids research 39, no. 8 (2011): 3017-3025.

[11] Leung, Michael KK, Andrew Delong, Babak Alipanahi, and Brendan J. Frey. "Machine learning in genomic medicine: a review of computational problems and data sets." Proceedings of the IEEE 104, no. 1 (2015): 176-197.

[12] Buck, Michael J., and Jason D. Lieb. "ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments." Genomics 83, no. 3 (2004): 349-360.

[13] Chen, Chin-Fu, Xin Feng, and Jack Szeto. "Identification of critical genes in microarray experiments by a Neuro-Fuzzy approach." Computational Biology and Chemistry 30, no. 5 (2006): 372-381.

[14] Chou, Chia-Cheng, Ting-Wan Lin, Chin-Yu Chen, and Andrew H-J. Wang. "Crystal structure of the hyperthermophilic archaeal DNA-binding protein Sso10b2 at a resolution of 1.85 Angstroms." Journal of bacteriology 185, no. 14 (2003): 4066-4073.

[15] Li, Qing, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. "Medical image classification with convolutional neural network." In 2014 13th international conference on control automation robotics & vision (ICARCV), pp. 844-848. IEEE, 2014.

[16] Siar, Masoumeh, and Mohammad Teshnehlab. "Brain tumor detection using deep neural network and machine learning algorithm." In 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 363-368. IEEE, 2019.

[17] Liu, Bingqiang, Ling Han, Xiangrong Liu, Jichang Wu, and Qin Ma. "Computational prediction of sigma-54 promoters in bacterial genomes by integrating motif finding and machine learning strategies." IEEE/ACM transactions on computational biology and bioinformatics 16, no. 4 (2018): 1211-1218.

[18] Kumar, Manish, Michael M. Gromiha, and Gajendra PS Raghava. "Identification of DNA-binding proteins using support vector machines and evolutionary profiles." BMC bioinformatics 8, no. 1 (2007): 1-10.

[19] Freeman, Katie, Marc Gwadz, and David Shore. "Molecular and genetic analysis of the toxic effect of RAP1 over expression in yeast." Genetics 141, no. 4 (1995): 1253-1262.