Hindawi
Applied Bionics and Biomechanics
Volume 2022, Article ID 5483115, 17 pages
https://doi.org/10.1155/2022/5483115

Research Article

DNAPred_Prot: Identification of DNA-Binding Proteins Using Composition- and Position-Based Features

Omar Barukab,[1] Yaser Daanial Khan,[2] Sher Afzal Khan,[3] and Kuo-Chen Chou[4]

[1]*Department of Information Technology, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, P. O. Box 344, Rabigh, 21911 Jeddah, Saudi Arabia*
[2]*Department of Computer Science, School of Systems and Technology, University of Management and Technology, P.O. Box 10033, C-II, Johar Town, Lahore 54770, Pakistan*
[3]*Department of Computer Sciences, Abdul Wali Khan University Mardan, Pakistan*
[4]*Gordon Life Science Institute, Boston, MA 02478, USA*

Correspondence should be addressed to Sher Afzal Khan; sher.afzal@awkum.edu.pk

Received 15 September 2021; Revised 25 December 2021; Accepted 5 February 2022; Published 13 April 2022

Academic Editor: Christian Maurer

In the domain of genome annotation, the identification of DNA-binding protein is one of the crucial challenges. DNA is considered a blueprint for the cell. It contained all necessary information for building and maintaining the trait of an organism. It is DNA, which makes a living thing, a living thing. Protein interaction with DNA performs an essential role in regulating DNA functions such as DNA repair, transcription, and regulation. Identification of these proteins is a crucial task for understanding the regulation of genes. Several methods have been developed to identify the binding sites of DNA and protein depending upon the structures and sequences, but they were costly and time-consuming. Therefore, we propose a methodology named "DNAPred_Prot", which uses various position and frequency-dependent features from protein sequences for efficient and effective prediction of DNA-binding proteins. Using testing techniques like 10-fold cross-validation and jackknife testing an accuracy of 94.95% and 95.11% was yielded, respectively. The results of SVM and ANN were also compared with those of a random forest classifier. The robustness of the proposed model was evaluated by using the independent dataset PDB186, and an accuracy of 91.47% was achieved by it. From these results, it can be predicted that the suggested methodology performs better than other extant methods for the identification of DNA-binding proteins.

1. Introduction

DNA (Deoxyribonucleic acid) is a blueprint for the cell. It contains information that is encoded for all our characteristics. A living thing's DNA is what makes a living thing a living thing. It is an essential part of reproduction that is transmitted from parents to offspring. There are four primary functions of DNA, commonly known as replication, encoding information, gene expression, and mutation and recombination. But DNA does not do this all alone; thousands of proteins in the cells help DNA to regulate DNA functions. Actions related to DNA are carried out with the help of specific proteins in living cells. These actions are carried out as the result of protein-DNA synergy [1]. Nonspecific or specific binding between DNA and protein is involved in achieving regulation. Proteins that attach to DNA for such governance are known as DNA-binding proteins. These DNA-binding proteins contain a domain of DNA-binding and have an affinity for single- as well as double-stranded DNA. At different stages of life, these functional proteins play a vital role [2].

Moreover, DNA-protein binding plays an imperative role in the gene study and the development of a living body. Their research also helps in an inspection of the human body. It helps in the identification of the procedure of actions taking place in the body such as ailment, growth, development, changes, and improvement.

In the development of cell and growth systems, an important role is played by the transcription factor. It usually resides in a cell with an inactive state, and the existence

of ligand TF becomes active. Desireless activation is responsible for many diseases such as inflammation, development disorder, autoimmunity, cancer, and abnormal hormone responses. Therefore, keeping a continuous record of DNA-binding proteins is of significant interest. It helps in the identification of, and treatment of diseases such as abnormal TF activity, cancers and genetic disorder which includes haemophilia, colour blindness, and many more. DNA-BP also plays an integral part in prokaryotic host defence in the shape of restriction enzymes. Binding of DNA with protein is shown in Figure 1.

Many experimental approaches used in biology have been adopted for the identification of DNA-binding proteins. These include X-ray crystallography [3], chromatin immunoprecipitation with DNA microarrays [4], and filter-binding assays [5]. These methods enable us to make exact identification of DNA-protein binding, but these mechanisms for proteins structures recognition are laborious, time-consuming, and require comprehensive material and expanse.

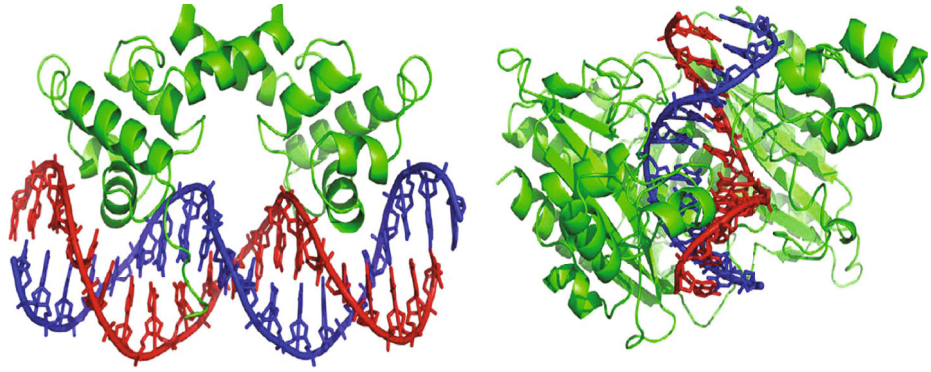
There are two practical approaches for the identification of sequences based on protein behaviour. One is the ML algorithms, to make improvements and expert model with derived numeral feature vector and query sequence forecasting. The second is the elicitation of organic information enclosed in the sequence of the protein and its metamorphosis into a comparable numeral vector of the features. Modern computational approaches for the identification of DNA binding protein are classified into two main classes: (1) Machine learning-based and (2) template-based.

Based on machine learning, DNA-binding protein prediction methodologies are divided into two general categories: structure-based [8, 9] and sequence-based [10–14] prediction. Higher identification rates can be achieved by the structure-based prediction of DNA-binding protein. Still, due to the inadequacy of sufficient knowledge about the structure of a protein, these approaches are not used on a large scale for the perception of high-throughput sequences. For predicting the function of a protein, new approaches are based on sequences of amino acids. By the result of bountiful experiments and methods, it realizes that proteins or primary polypeptide structure resembles the structural arrangement of polypeptide after wrapping and their methods are also very identical [15]. Template-based methods are also known as a template-based methodology because this identifies the consequential correspondence of protein sequences or structure among a known template and a query to bind DNA, to determine and evaluate the DNA-binding priority of sequences that are targeted [16, 17]. In contrary to the template-based approach, machine learning methodology determines a similar forecasting model to predict by analyzing and identifying the arrangement and pattern in feature space input. Some cases are support vector machine (SVM) [11, 12, 18–20], random forest [21], neural network [22–25], nearest neighbors' algorithm [23], naïve Bayes classifier [26, 27], and ensemble classifiers [28–30]. The process of identifying DNA binding protein by utilizing machine learning techniques requires two essential steps: (1) compatible feature extraction and (2) selection of suitable classification algorithm. The extant predictive methodology can be divided into two sections based on feature elicitation methods: (1) from protein structure extract appropriate features [31–34] and (2) relevant feature extraction from amino acid sequences [8, 35–38]. For DNA-binding protein recognition, more accurate and authentic results can be obtained using a structure-based prophecy technique [39]. Still, for this, a 3D structure with a high resolution of the protein sequence is required.

Thus, until now, for the identification of DNA-binding protein, many computing techniques direct from their amino acid sequences have been proposed and suggested. These approaches independently analyze and probe four distinct kinds of a feature of protein sequences and ciphering sequences [11, 39–42]. Categorically, the four specific types consist of (1) structural information, (2) functional and compositional information, (3) information about evolution, and (4) physicochemical properties. The four distinct categories of encoding procedures are as follows: (i) OCTD (global strategy) overall composition-transition-distribution, (ii) SSA transformation (local procedure) called split amino acid, (iii) ACC transformation (nonlocal approach) autocross covariance, and (iv) position-specific scoring matrix distant transformation known as "PSSM-DT". These procedures have been considered deep in their related scrutinize work [28, 39, 43, 44].

There exist few recent studies which perform prediction of DNA-binding proteins using multiple features and machine learning classifiers. In 2022, Zhang et al. proposed a novel method for prediction of DNA-binding proteins by using features from amino acid composition and evolutionary information of protein sequences. Later, these features were fed to an XGboost classifier [45]. Furthermore, Harini et al. in 2022 created a database named ProNAB for DNA and protein complexes [46]. Jia et al., in 2021, proposed KKDBP, a classifier for the prediction of DNA-binding proteins using multiple PSSM feature fusions and random forest as a classifier [47]. In 2021, Hu et al. proposed TargetDBP+, which performed prediction of DNA-binding proteins using five convolutional features and SVM classifier [48]. Qian et al. in 2021, extracted six sequence-based features and used Multiple Kernel Learning-based on Centered Kernel Alignment for fusion of these features. Further, SVM was used for the classification of DNA-binding proteins [49]. Zou et al. proposed FTWSVM-SR, which used multiple sequence-based features and SVM as a classifier for predicting DNA-binding proteins [50]. Zou et al. also proposed MK-FSVM-SVDD, another predictor for DNA-binding protein prediction using six features with central kernel alignment and SVM as classifier [51]. However, the accuracy of all these proposed methods still has room for improvement. Nevertheless, most of the suggested approaches are inadequate in their capability to describe protein-DNA binding. Therefore, it is vital to develop a new strategy for the prediction of DNA-binding proteins accurately and efficiently and to compare it with existing state-of-the-art techniques.

The present work focuses on the identification of DNA-binding proteins through sequences. There are usually two goals for predicting DNA-binding proteins with different



(a) (b)

Figure 1: DNA binding protein bound to respective target DNAs. Created from PDB (a) 1LMB and (b) 1 RVA. Image source [6, 7], respectively.

techniques: (1) to help scientists for the development and get covet data and (2) to encourage academic studies for appropriate fields. For establishing a sound analytical protein identification system, we need to deal with following the 5-step rule that includes (a) a valid standard dataset, (b) sample formulation, (c) algorithm for operation purpose, (d) performing cross-validation, and (e) friendly user web server for forecasting which is publicly accessible. The proposed system is highly accurate as compared to the previously existing methods and is easy to opt for as it only uses sequence-based features of proteins to identify them as DNA binding or non-DNA binding.

2. Materials and Methodology

The methodology is divided into five steps, the first aspect, which is "A valid benchmark," is discourse here in this section. The protein sequence benchmark dataset was obtained from UniProtKB. At first, all types of sequences are passed out from a process of CD-HIT, which stands for Cluster Database at High Identity with Tolerance, and is initially composed by Weizhong Li and is now available publicly. The basic functionality of CD-HIT is to take input in FASTA format and remove similar or highly similar sequences from the dataset. The purpose is to reduce the size of the dataset by removing redundant or highly matching sequences from the dataset. So, for a benchmark dataset used in this study, sequences' identity cut-off is set to 60%. Redundant sequences or 60% identical were removed out, and a dataset is formed. All sequences of the obtained dataset are classified into two categories: (a) positive and (b) negative. These sequences of the DNA-binding protein are available in the dataset named "Dataset". The dataset contains 57,194 DNA-binding protein sequences in which there are positive 11,526 sequences. Moreover, to check the robustness of the proposed methodology model, an independent dataset PDB 186 [40] has also been used. There are 93 binding proteins and 93 nonbinding protein sequences in an independent dataset. The performance of the proposed method has been compared with state-of-the-art methodologies. The details of datasets are shown in Table 1.

For the identification of DNA-binding protein, the methodology followed includes data collection from UniProt, applying preprocessing and filtration techniques, after that calculating the features obtained, in the end, training the classifier and getting the results, as shown in Figure 2.

2.1. Extracting Features. The second step describes how the dataset samples are devised into proper expressions of mathematics which equate and compare these samplings with aimed biological class in a remarkably precise, efficient, and accurate way.

Such a formulation of samples is essential depending upon the static nature of classifiers. With frenzied extension and expansion of biological sequences in a postgenomic era, one of the most complex and critical issues in bioinformatics is to identify the suitable way to define these sequences with vectors based on unique models. Such notations and transformations assist in maintaining the unique arrangement of sequence characteristics and essential information about proteomic data. Machine learning algorithms are incorporated to use vectors for entertaining them, but a dataset of sequences needs to decipher among classes based on data extracted by the transformation process [52]. There is a risk that a vector which is represented in a discrete structure may mislay information about sequences completely or to bypass from complete loss of information of sequences arrangement for protein, a strategy named 'PseAAC' [53] was suggested which stands for the "Pseudo Amino Acid Composition" [54]. This strategy has been prevalently used in all fields of proteomic calculation [55–61]. This extensive and progressive use led to the formation of three existing opened access powerful and useful softwares, called "PseAAC-Builder", "propy", and "PseAAC-General", for developing different methods of Chou's special PseAAC [62] where the last one is a generalization of "PseAAC" [63]. They not only include the distinctive approach for feature extraction of proteomic data but also extend to feature vectors which include, "Functional Domain" mode, "Gene Ontology" mode, and "Sequential Evolution" or "PSSM" mode. Inspired by the complementary outcome of utilizing "PseAAC" to handle the sequences of peptide or protein, the proposed strategy of "PseAAC" was continued to Pseudo K-tuple Nucleotide Composition (PseKNC) for

Table 1: Detail of the dataset used.

Sequences	Benchmark dataset	Independent dataset
Negative sequences	45,668	93
Positive sequences	11,526	93
Total	57,194	186

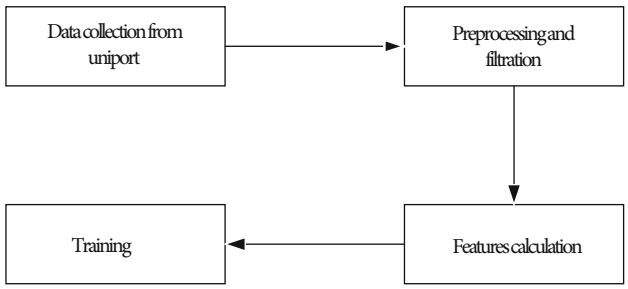


Figure 2: Flowchart of the proposed methodology.

developing and achieving different feature vectors for RNA/DNA that have confirmed very favourable as well [64–70]. Especially, recently, an advanced web server named "Pse-in-One" [71] and "Pse-in-One 2.0" [72], which is its advanced version and can be utilized in generating any required protein/peptide vector and sequences of DNA and RNA according to the requirement of the users. Here are some methodologies used for extracting the features, to identify the specific arrangements associated with the primary protein structure.

2.2. Position Relative Incidence Matrix (PRIM). The first step is to transform the primary structure of protein into a matrix form for expressing the typical features of proteins. PRIM is built by utilizing the protein sequence length. With the help of a row-major strategy, protein basic structure is converted into two-dimensional from singular dimensional. We can calculate the two-dimensional matrix by the following equation if we simply take the square root of the length of the protein.

$$n = \left|\sqrt{k}\right|, \tag{1}$$

Here, n and k are the two-dimensional square matrix dimension and primary sequence length, respectively. Later on, this amino acid matrix is used in the computation of PRIM through which the development of feature vector is done. The formation structure of PRIM is 20x20. The representation of two dimensional is as follows in equation (2).

$$S_{PRIM} = \begin{bmatrix} Y_1 \longrightarrow 1 & Y_1 \longrightarrow 2 & \cdots & Y_1 \longrightarrow j & Y_1 \longrightarrow 20 \\ Y_2 \longrightarrow 1 & Y_2 \longrightarrow 2 & \cdots & Y_2 \longrightarrow j & Y_2 \longrightarrow 20 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ Y_i \longrightarrow 1 & Y_i \longrightarrow 2 & \cdots & Y_i \longrightarrow j & Y_i \longrightarrow 20 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ Y_n \longrightarrow 1 & Y_n \longrightarrow 2 & \cdots & Y_n \longrightarrow j & Y_n \longrightarrow 20 \end{bmatrix}. \tag{2}$$

Here, Y figures out the i^{th} position residue score relative to j^{th} type amino acid. The possible values for j could be 0, 1, 2, 3, 4…, and so on. This 20x20 matrix can produce a total of four hundred coefficients. Statistical moments are computed for PRIM by reducing the number of coefficient elements which is 24 in the case of PRIM computation. 10 raw, Hahn, and central moments were calculated up to order three, and hence, 30 unique features were obtained.

2.3. Reverse Position Relative Incidence Matrix (RPRIM). To explore concealed and complicated characteristics of an elementary sequence of the protein that has confusion with similar sequences of other protein, a matrix is used which have 400 coefficients as it contains 20x20 dimension as PRIM, known as reverse position incidence matrix.

$$S_{PRIM} = \begin{bmatrix} Y_1 \longrightarrow 1 & Y_1 \longrightarrow 2 & \cdots & Y_1 \longrightarrow j & Y_1 \longrightarrow 20 \\ Y_2 \longrightarrow 1 & Y_2 \longrightarrow 2 & \cdots & Y_2 \longrightarrow j & Y_2 \longrightarrow 20 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ Y_i \longrightarrow 1 & Y_i \longrightarrow 2 & \cdots & Y_i \longrightarrow j & Y_i \longrightarrow 20 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ Y_n \longrightarrow 1 & Y_n \longrightarrow 2 & \cdots & Y_n \longrightarrow j & Y_n \longrightarrow 20 \end{bmatrix}. \tag{3}$$

Dimensions of the matrix mentioned above are reduced. Statistical moments are calculated for RPRIM, which have 24 elements set. 10 raw, Hahn, and central moments are calculated using 2D S_{RPRIM} up till third order, 30 unique features obtained.

2.4. Statistical Moments. In recognition of patterns, many research methodologies demonstrate that statistical moments are fruitful to generate features against those sequences which do not rely upon any guideline. A specific category of biased average, which is used in analyzing the consolidation of some unique structure in problems related to sequence recognition is known as moments [73]. These are also helpful in many issues related to pattern recognition. Another important method for determining and understanding different kinds of sequences and object depiction is orthogonal moments.

By using techniques of polynomial and distribution functions, many statisticians develop certain moments. Further, Hahn, central, and raw moments are utilized to explain the problem in discussing in this study. There are two types of orthogonal moments, (1) discrete moments and (2) continuous moments. It has been considered in a recent study [74] that for quantized and distinct data, the result gained by a discrete moment was much better than a continuous moment. A different form of the moment can be calculated by the matrix or vector collection which represents any pattern. The raw moments are treated as generally known moments which can be calculated using the below equation (4).

$$M_{xy} = \sum_i \sum_j i^x j^y f(i,j). \tag{4}$$

The origin of data is considered as a remark point by the raw moments; on the other hand, components that are far away from the origin point are used in calculating the moments. The data's centroid is used by central moments as their remark point, which was calculated by the following equation (5).

$$U_{xy} = \sum_{p}\sum_{q}\left(p - p'\right)pow\,x\left(q - q'\right)pow\,yf(p,q). \tag{5}$$

Distinct features up to third order are obtained with the help of central moments and defined as U_{00}, U_{01}, U_{10}, U_{11}, U_{02}, U_{20}, U_{12}, U_{21}, $U_{30,}$ and U_{03}. Now, the centroids p' and q' are computed from equations (11) and (13).

$$\begin{aligned} p' &= \frac{\text{the}\,M^{10}}{M^{00}}, \\ q' &= \frac{M^{01}}{M^{00}}. \end{aligned} \tag{6}$$

Orthogonal moments which need a square matrix input data in two-dimensional are Hahn moments of two dimensional. They can be calculated when the notations of one-dimension are converted into square matrix notations. N order of Hahn polynomial is calculated from the Eq. (7).

$$h_n^{u,v}(r,n) = (N+1+r)_n(N-1)_n, \tag{7}$$

$$\sum_{\mathrm{k}=0}^{\mathrm{n}} (-1)^{\mathrm{k}} \frac{(-\mathrm{n})_{\mathrm{k}}(-\mathrm{r})_{\mathrm{k}}(2\mathrm{N}+\mu+\mathrm{v}-\mathrm{n}-1)_{\mathrm{k}}}{(\mathrm{N}+\mathrm{v}-1)_{\mathrm{k}}(\mathrm{N}-1)_{\mathrm{k}}} * \frac{1}{\mathrm{k!}}. \tag{8}$$

Generalization of the Pochhammer symbol is made as in equation (9).

$$(a)_k = a(a+1)\cdots.(a+k-1). \tag{9}$$

The Pochhammer symbol will become more simplified when using an operator named Gamma as follows in equation (17)

$$(a)_k = \frac{\Gamma(a+k)}{\Gamma(a)}. \tag{10}$$

Raw values for Hahn's moments are generally measured by utilizing a square norm and weighting method, as shown in Eq. (22).

$$h_n^{\dot{v},v}(r,N) = \sqrt{\frac{\rho(r)}{d_n^2}}, n = 0, 1, \cdots N-1. \tag{11}$$

On the other hand, in equation (12).

$$\rho(r) = \frac{\Gamma(r+\mu+v) + \Gamma(r+v+1)(r+1+\mu+v)_N}{(v+\mu+2r+1)n!(N-r-1)!}. \tag{12}$$

The Hahn moments which are orthogonally normalized for discrete data of two dimensional are calculated up to three orders as mentioned in equation (13).

$$H_{ij} = \sum_{q=0}^{N-1}\sum_{p=0}^{N-1} \beta_{pq} h_i^{\dot{v},v}(q,N) h_j^{\dot{v},v}(p,N), m, n = 0, 1, 2 \cdots N-1. \tag{13}$$

For every sequence 10 raw, 10 central, and 10 Hahn moments are calculated up to third order. Features obtained by Hahn moment are represented as H_{00}, H_{01}, H_{10}, H_{11}, H_{02}, H_{20}, H_{12}, H_{21}, H_{30}, and H_{03}. By using the methods mentioned above, we can obtain feature vectors, after that, they are used in training and in developing a classifier.

2.5. Frequency Vector. In sequences, the number of the existence of amino acid is represented by frequency; a vector is figured out for frequency distribution measurement known as frequency vector.

$$\xi = \{\tau_1 + \tau_2 + \tau_3, \cdots, \tau_{20}\}. \tag{14}$$

Here, in the above equation, the occurrence frequency of an amino acid i^{th} residue is denoted by τ_i. The primary purpose of calculating this vector is to uncover and reveal the hidden sequence compositional information. A total of 20 unique features were obtained that were used with others for training purposes.

2.6. Accumulative Absolute Position Incidence Vector Formation (AAPIV). The purpose of the frequency matrix is to obtain compositional information about the sequence. Still, the knowledge about the residue relative position did not get from it, for this purpose, a vector named accumulative absolute position incident is computed, which has a length of 20 elements. In this vector, the mean of all statistical values for every endemic amino acid, appearing in a primary sequence is located at their specific locations, and 20 features are obtained from it.

This vector can be denoted as M and represented in equation (15):

$$M = \{\mu_{1,}\mu_{2,}\mu_{3,}\cdots.,\mu_{20+}\}. \tag{15}$$

For the computation of i^{th} arbitrary AAPIV's element, below mentioned equation is used.

$$\mu_i = \sum_{M=1}^{n} P_M. \tag{16}$$

2.7. Reverse Accumulative Absolute Position Incidence Vector (RAAPIV). RAAPIV is generated by overturning the primary sequence and producing the AAPIV from the overturn sequence. Hence, give 20 unique features. The primary purpose of developing RAAPIV is to draw out and uncover the facts and figures from the relative residue's position of the sequences. This reverse vector is represented as

$$\Lambda = \{\eta_{1,}\eta_{2,}\eta_{3,\cdots,}\eta_{20}\}. \tag{17}$$

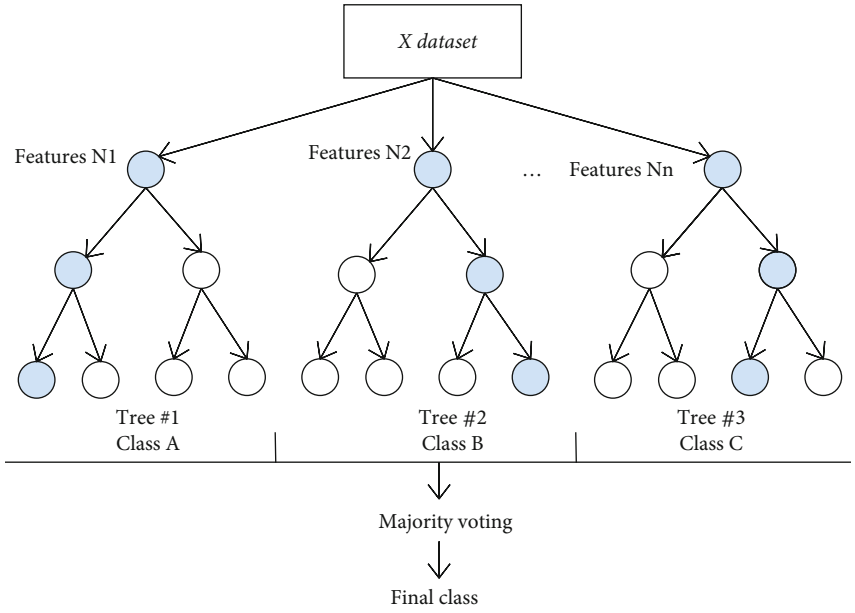


Figure 3: Representation of random forest classifier.

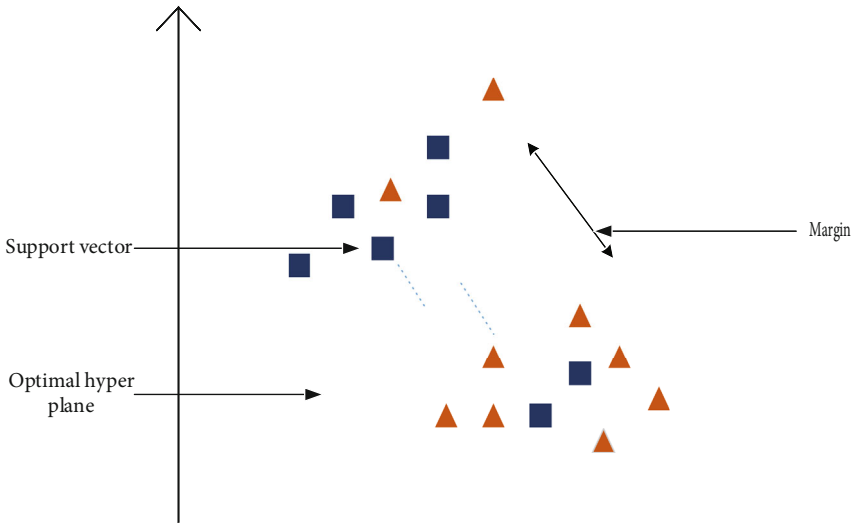


Figure 4: Representation of support vector machine.

2.8. Feature Fusion. After passing through all the procedures mentioned above, multiple features were fused into one vector. PRIM and RPRIM were converted into concise data by calculating moments (such as raw, central, and Hahn) and further integrated into a feature vector as well as with AAPIV and RAAPIV. This yielded 100 features. All these features helped in defining relative positions as well as absolute positions of amino acid residues. Furthermore, frequency-based features were computed through frequency vector, which elaborated the frequency of amino acids and yielded 20 features.

2.9. Algorithms for Classification. The third stage of the five-step rules of Chou's is elaborated in this part, which is the formation of an operational algorithm. For classification, one of the most commonly used methodologies, Random Forest (RF) has been adopted at this stage. To compare results from the random forest "Support Vector Machine" (SVM) and "Artificial Neural Network" (ANN) were also used. In research studies related to bioinformatics, methods of ensemble learnings have been practiced [74, 75] and efficient results produced by them in terms of performance. In ensemble learning techniques, the results of all several classifiers used for solving particular problems are aggregate. The two most frequently used schemes are bagging [76] and boosting [77].

Bagging the trees which are succeeding to the previous does not depend upon the preceding trees; instead, each tree is formulated independently utilizing a bootstrap sample from the data available. In the end, the prediction is

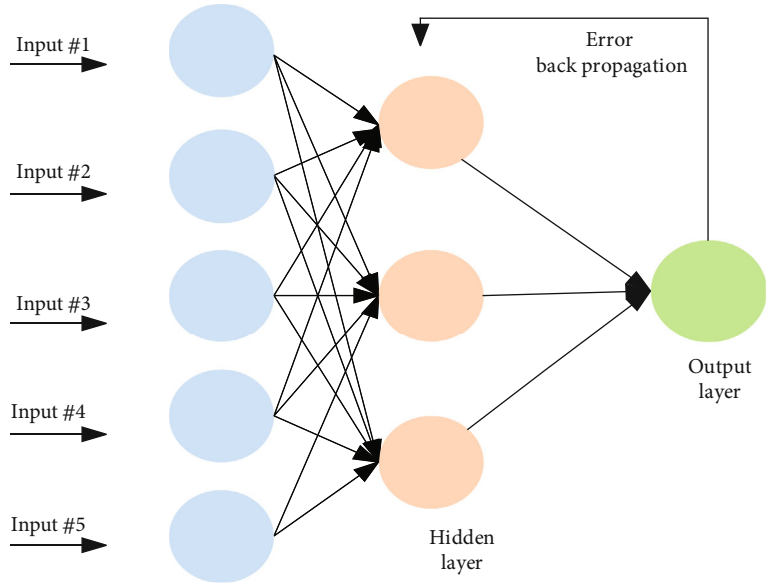


Figure 5: Artificial neural networks working representation.

Table 2: Description of equation symbols.

Symbols	Description
N^+	The total number of true DNA-binding proteins
N_-^+	The total number of true DNA-binding proteins incorrectly predicted as nonbinding proteins
N^-	Total number of true non-DNA-binding proteins
N_+^-	The total number of true non-DNA-binding proteins incorrectly identified as DNA-binding proteins

Table 3: Description of possible values.

When,	Then,	Details
$N_-^+ = 0$	Sn = 1	None of the DNA-binding proteins is predicted as non-DNA-binding protein
$N_-^+ = N^+$	Sn = 0	All of the DNA-binding protein is incorrectly predicted as non-DNA-binding protein
$N_-^+ = 0$	Sp = 1	None of the non-DNA-binding proteins is incorrectly predicted as DNA-binding protein.
$N_+^- = N^-$	Sp = 0	All of the non-DNA-binding proteins incorrectly predicted as DNA-binding proteins
$N_-^+ + N_+^- = 0$	MCC = 1, ACC = 1	None of the DNA-binding protein and none of non-DNA-binding protein was incorrectly predicted
$N_-^+ = N^+$ and $N_+^- = N^-$	MCC = −1, ACC = 0	All of the DNA-binding protein and all of non-DNA-binding protein was incorrectly predicted
$N_-^+ = (N^+/2)$ and $N_+^- = N^-/2$	ACC = 0.5, MCC = 0	Overall prediction is not good enough than any other random prediction outcomes.

Table 4: 10-fold cross-validation results.

Classifier	True positive	False positive	True negative	False negative	Accuracy
Random Forest	45,480	2,748	8,778	127	94.97%
Artificial neural network	45,540	11,404	112	67	79.5%
Support vector machine	21,529	5,314	6,212	24,078	48.55%

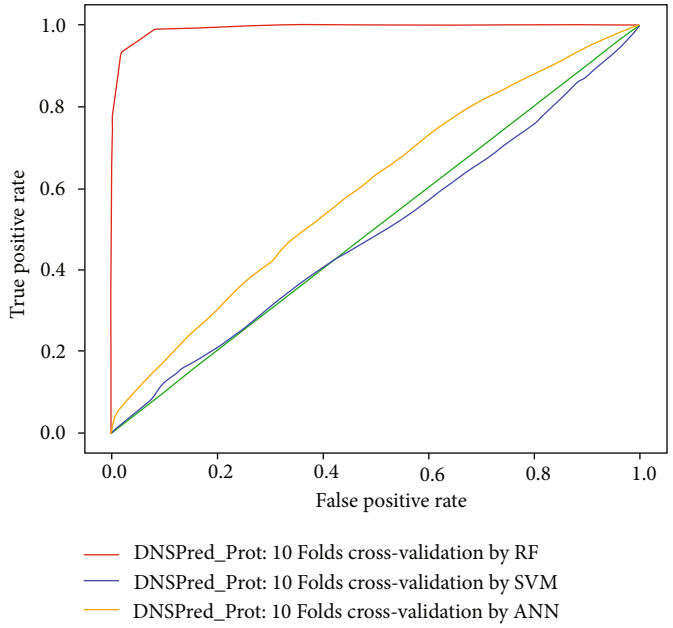


Figure 6: ROC comparison for 10-fold cross-validation.

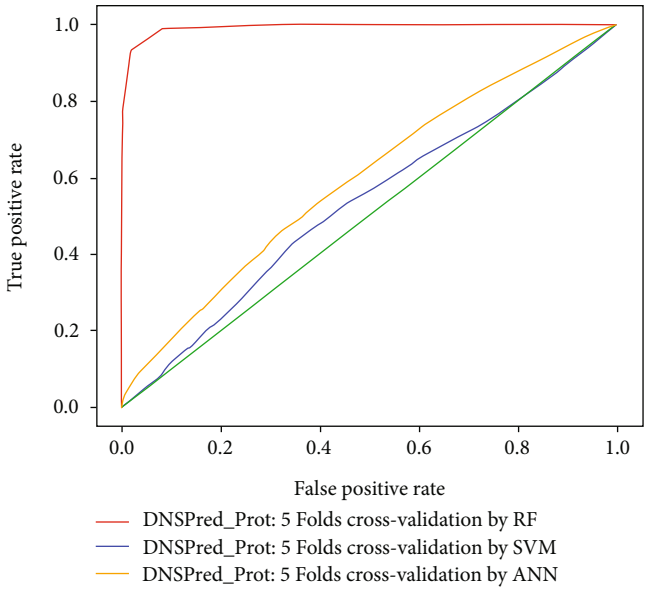


Figure 7: ROC comparison for 5-fold cross-validation.

determined by a simple ballot majority. Contrary to this, trees that are next in order in boosting promulgate additional value to points that were incorrectly anticipated by a former classifier. In the end, the weighted majority determines the prediction. Random forest is built by Adele Cutler and Leo Breiman [6]. A supplementary layer of randomness is an add-on to bagging. Usually, in classification trees, the partition of each node is performed by distributing a node equally between all available variables, whereas in random forest, splitting is done by selecting perfect among the available predictor's subset which was selected arbitrary were at that node. The random forest becomes a counterintuitive approach that is firmly against overfitting and performs effectively.

Random forest is an ensemble of decision trees where the training (sample) dataset is recursively partitioned into different decision trees based on the value of a parameter. It is firmly across overfitting, fast, and scalable, which enables it to give better results with an increasing number of examples.

A random forest is also known as a random decision forest because at the time of training, tasks are operated by making a multitude of decision trees, and at the time of output, the class which is the mode of all the classes used in the process or individual trees mean evaluation is given as the final result. A pictorial representation of the random forest is shown in Figure 3.

In machine learning, SVM is a supervised machine learning model. These are selective classifiers that are formally designed by a separable hyperplane. Initially, it is introduced in the 1960s and improved in the 1990s. Its working in space example can be easily understood by points. Points of each category are separated. In case the gap between an instance of different types is more massive, more comfortable to identify the cluster. So, the primary purpose of SVM is to segregate the available data in the best possible way. For this purpose, SVM kernels are used; their primary function is to add more dimensions to low dimension space. By using the kernel, an inseparable problem can be converted to a separable problem. SVM is always implemented and practiced by the kernel. Some types of the kernel are as follow: (a) linear kernel, (b) polynomial kernel, and (c) radial basis function kernel. The main advantage of SVM is that it works well in cases where the number of dimensions is greater than the number of samples. It also performs well when the space between classes is large. It does not perform well when the available data is too large

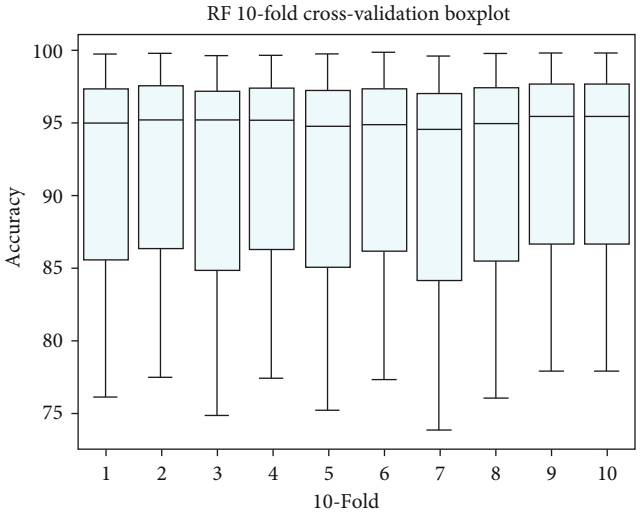


Figure 8: Box plot for Random Forest.

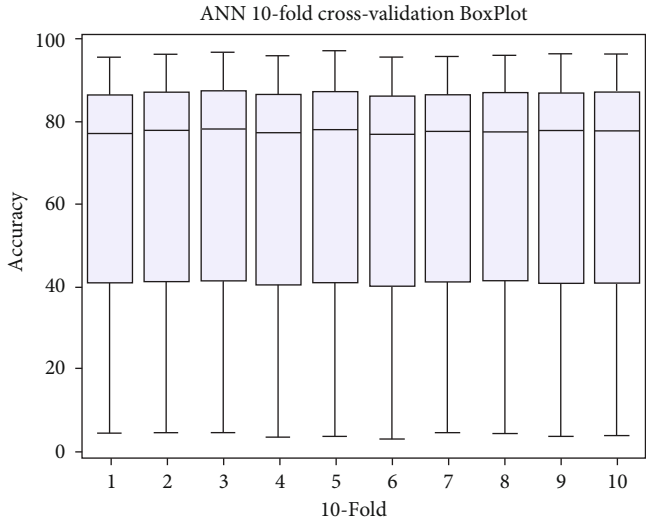


Figure 9: Box plot representation for ANN.

or contains too much noise. SVM was used in this study just to compare results with a random forest of cross-validation, jackknife, self-consistency, and independent testing to check the effectiveness and validity of random forest. The working of SVM can be seen in Figure 4.

The processing way of the brain is adopted as a foundation for an artificial neural network. It falls in the category of supervised learning technique which utilizes backpropagation to train data. ANN is used in solving a vast dimension of the problem. It can easily discriminate nonlinear data. ANN is a framework of coupled neurons in which the next neuron input is the output of the previous one, as shown in Figure 5. A connection is known as an edge, both edge and neuron's weight help in the learning process. In ANN, outputs of the previous neuron become the input of the next neuron. The following equation represents ANN working.

$$O_m = f\left(\sum_{b=1}^{h} W_{bn} * f\left(\sum_{a=1}^{i} W_{ab} X_a\right)\right). \tag{18}$$

Here in the above equation, the input is represented by i, the total number of output nodes and hidden layer nodes are represented by o and h, respectively. O_{m} denotes every m^{th} neuron output. X_{a} acts as an input for node a. The weight of edge connecting node a and of input layer to node b of the hidden layer is denoted by $W_{\mathrm{ab},}$ whereas the weight of connecting output layer node to node b is represented by W_{bn}. At last, the neuron activation function is a classical sigmoid function that is denoted as f.

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{19}$$

The prior formulated benchmark dataset contains positive as well as negative samples. For all collected models, a feature vector is calculated against each of them. Every feature vector consists of Hahn, raw, and central moments of the basic structure of protein for two-dimensional depiction, RPRIM along with PRIM. Furthermore, information about the position and composition is obtained in the form of the Frequency Matrix (FM). By associating all the feature vector, so each row correlates to a unique individual specimen and forms a Feature Input Matrix (FIM). Then, a matrix is acquired in an administrative aspect that adjusts to the category, i.e., negative or positive of the equivalent component in the Frequency Input Matrix. These matrices which have been discussed before, are used in training of the random forest, support vector machine and artificial neural network [75].

2.10. Adaptive Learning and Gradient Descent. In the training of an algorithm, gradient descent is used. This reduces the motion of the function in the contradictory route of the function's gradient and change in the rate is calculated in a further output such that

$$\theta = \theta - \gamma \nabla_{\varnothing} F(\theta), \tag{20}$$

where theta θ is a parameter to the objective function F, θ is an element of d, the learning rate which is shown by γ, and the gradient function is represented as $\nabla_{\theta} F(\theta)$. The overall algorithm efficiency depends upon the rate of learning γ because it ascertains the effective minimization.

There should be optimal values for the learning rate, and it is kept small, usually because more time is taken by a small percentage to join. The convergence, on the other hand, function oscillation may be caused due to the large learning rate. An adaptive learning algorithm calculates fluctuation in the learning rate and it depends on algorithm performance. On comparing the two consecutive iteration errors if an error in second as to first increases, then parameters used for that particular iteration are dismissed and the rate of learning fluctuates in a specific manner that function is downplayed by it. By usage of two consecutively calculated parameters, the weights used are again computed, and as a result, the output is also recomputed. For that ensuing run consequent errors that may occur are also calculated. Finally,

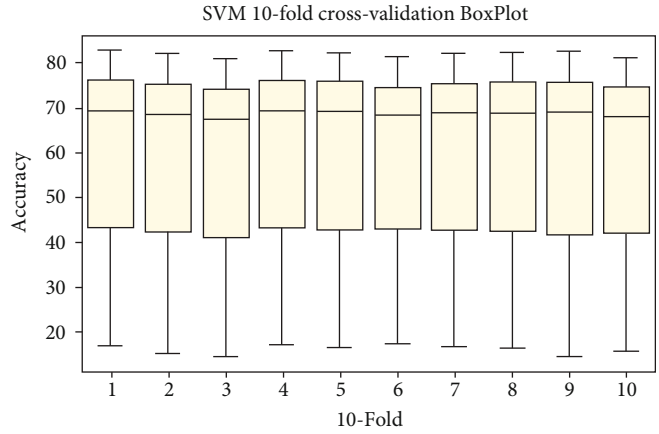


Figure 10: Box plot representation for SVM.

Table 5: Accuracy obtained from jackknife testing.

Classifier	True positive	True negative	False positive	False negative	Accuracy
Random Forest	45,492	8850	2,676	115	95.11%
Artificial neural network	45,540	11,404	112	67	79.5%
Support vector machine	21,529	5,314	6,212	24,078	48.55%

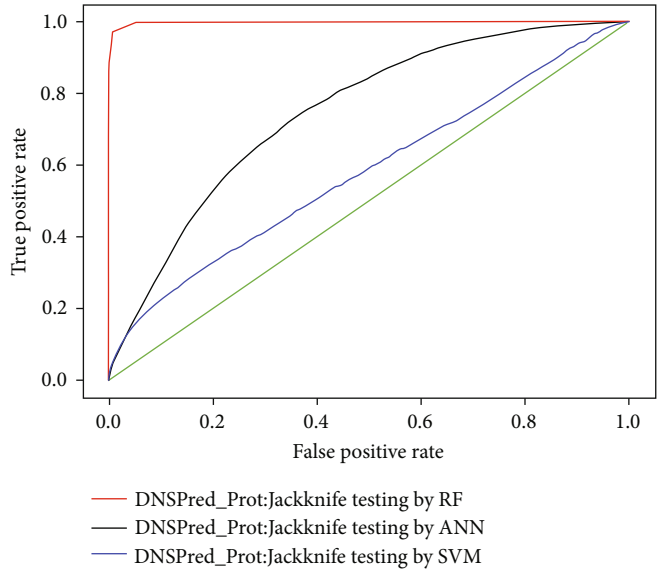


Figure 11: ROC for jackknife testing.

Table 6: Confusion matrix obtained from independent testing.

Classifier	Type	True positive	True negative	False positive	False negative	Accuracy
Random Forest	Testing	13,693	3,044	450	7	97.33%
Support vector machine	Testing	95	3,483	11	13,551	20.88%
Artificial neural network	Testing	13,646	0	3,494	0	79.61%

on comparing with a previously calculated error rate, if it is greater than the rate of learning is diminished, furthermore, the unique rate of theta +1 is calculated and weights are eliminated as well. Likewise, the learning rate becomes high for a nominal error rate. Hence, learning rate continuously varies depending upon the execution of an algorithm.

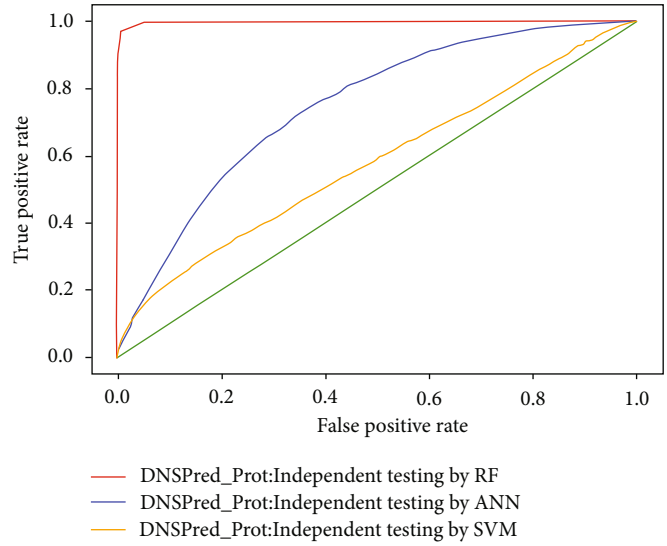


Figure 12: ROC of independent testing by classifiers.

Table 7: Accuracy obtained by self-consistency.

Classifier	True positive	True negative	False positive	False negative	Accuracy
Random Forest	45,474	8,859	2,667	133	95.1%
Artificial neural network	45,365	23	11,503	242	79.44%
Support vector machine	6212	5314	24078	21529	48.56

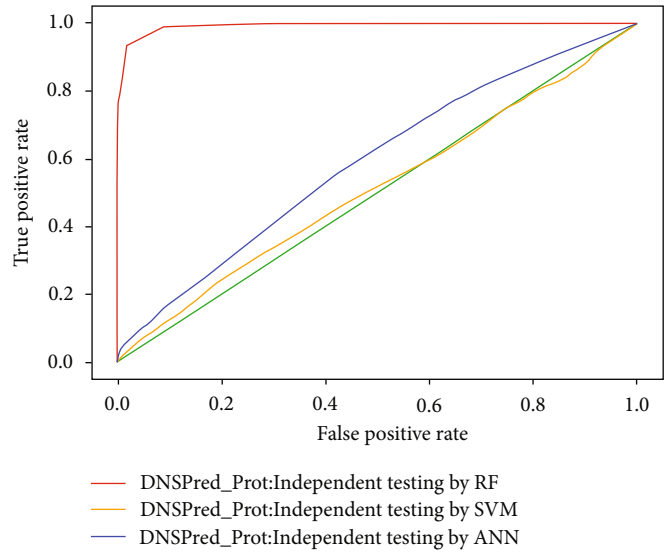


Figure 13: ROC comparison for self-consistency.

It is observed that the learning rate can fluctuate on each point and for a parameter of each succeeding epoch these are computed as follows:

$$\theta_{n+1} = \theta_n - \gamma_n \nabla F(\theta_n). \tag{21}$$

For the n^{th} epoch γ_n is the learning rate.

3. Experiment and Results

3.1. Prediction of Accuracy. Among many hurdles, one of the most substantial tasks in making a state-of-the-art prediction model is how the predicted model determines the rate of success objectively [58]. Focusing on this point, the proposed model requires two significant issues to examine. (1)

To quantitatively express the predictor capacity and excellence, which benchmark should be used? (2) What type of test procedure is used to explore and evaluate metrics? Several parameters with different techniques for all three classifiers were used to measure the performance.

3.2. Test Methodology. It is essential to consider which type of test methodology should be used to examine and rate the four metrics mentioned in Eq. (2). In the examination and determination of statistics, the coming three methods are commonly utilized in the inspection and analysis of the predictor.

(1) "Subsampling" (cross-validation) test, (2) "Jackknife Test" [71], and (3) "Independent dataset test" (IDT). Out of previously mentioned testing techniques, the one which is assumed the minimum inconsistent is jackknife. Jackknife produces the slightest different output for a given dataset on testing, explained in detail in the citation [58]

In case while confirmation set is not available, for establishing an exception that the methodology that was proposed is working excellent, the cross-validation technique is used. Dataset is divided into disassociate k-folds in cross-validation, while k is preserved fixed. For each partition obtained, testing is performed k-times on it after computed models for every single iteration training and accuracy. In the end, the absolute accuracy mean obtained is the outcome of the subsampling testing technique cross-validation. In the current scenario to get the result, k-fold cross-validation has been implemented, and an arbitrary choice to generate subsets for $k = 10$ was executed.

3.3. Formulation of Metrics and Evaluation Parameters. Presented metrics in Eq. (23) are commonly utilized to calculate prediction's degree of excellence from four different perspectives: (a) MCC for strength and stability, (b) Acc for measuring the precision and accuracy, (c) Sp for predictor specificity, and (d) Sn for the sensitivity of the predictor [74]. Regrettably, the traditional formulation of the abovementioned was provided in [76], most experienced scientists observe difficulties in understanding them, for MCC, it is especially. Amazingly, by using Chou's letter presented in analyzing peptide signals [77] Chen et al. [6] and Xu et al. [7] transformed them into a group of four intuitive equations, which are given as follows:

$$\begin{cases} Sn = \dfrac{TP}{TP + FN} \\ \dfrac{Sp = TN/TN + FP}{Accuracy = TP + TN/TP + FP + FN + TN} \\ MCC = \dfrac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + \mathrm{FP})(TN + FN)}} \end{cases} . \tag{22}$$

A symbol used for the conventional equation was introduced in ref. To define the equation, N_+^-, N_-^+, N^+, and N^- symbols were used. Their details are available in Table 2.

Table 8: Comparison of jackknife results with state-of-the-art predictors.

Metric/method	ACC	Sensitivity	Specificity	MCC
iDNA-Prot	0.7540	0.8381	0.6473	0.5000
PSSM-DT	0.7996	0.8191	0.7800	0.6220
DNA-binder	0.7358	0.6647	0.8036	0.4700
DNA-Prot	0.7255	0.8267	0.5976	0.4400
StackDPPred	0.8996	0.9112	0.8880	0.7990
DNAPred_Prot	**0.9511**	**0.9975**	**0.7678**	**0.8444**

Substitute symbols of Table 2 to Eq. (22) we get Eq. (23)

$$\begin{aligned} Sn &= 1 - \frac{N_-^+}{N^+} \\ Acc &= 1 - \frac{N_-^+ + N_+^-}{N^+ + N^-} \\ Sp &= 1 - \frac{N_+^-}{N^-} \\ MCC &= \frac{1 - ((N_+^-/N^-) + N_-^+/N^+)}{\sqrt{(1 + N_-^+ + N_+^-/N^-)(1 + N_+^- + N_-^+/N^+)}} \end{aligned} . \tag{23}$$

Eq. (21) and Eq.(20) have the same meaning but it becomes easy to understand what that equation means. Eq. (21) description is available in Table 3.

Thus, by equation (23), the overall accuracy, specificity, sensitivity, and MCC can be easily understood compared to the equation defined in (22) which is authenticated only forsingle-label systems. A real unique metric set is required for systems that are multilabelled as described in [78] and whose emergence is becoming common in biomedicine [79], system medicine [80], and system biology [81].

4. Discussion

Here, it is vital before going into the result section, to discuss the techniques used to get these results. As mentioned above, there are usually three popular testing techniques, (1) 10-fold cross-validation, (2) independent testing, (3) jackknife testing, and (4) self-consistency were used to validate the accuracy of the predictor model. So, in DNAPred_Prot, all the techniques were used to examine the accuracy of the proposed model. The classifier used in testing and training of the model was "Random Forest", "Support Vector Machine", and "Artificial Neural Network".

The accuracy achieved by DNAPred_Prot for the prediction of DNA binding proteins is better than models [14, 40] proposed previously. DNAPred_Prot results achieved can also be viewed in graphical representation; moreover, receiver operation characteristic curves for each testing technique were also done for more precise and efficient analysis. In the end, the web server was developed using a flask framework. It was done by following the five-step rule to facilitate others with these findings.

Table 9: Comparison of independent dataset PDB186 on the proposed method with other predictors.

Method	ACC	Sensitivity	Specificity	MCC
PSSM-DT	0.8000	0.8709	0.7283	0.6470
iDNA-Prot	0.6720	0.6770	0.6670	0.8330
DNA-Prot	0.6180	0.6990	0.5380	0.2400
DNAbinder	0.6080	0.6990	0.5380	0.2400
DNA-BIND	0.6770	0.6670	0.6880	0.3550
DBPPred	0.7690	0.7960	0.7420	0.5380
StackDPPred	0.8655	0.9247	0.8064	0.7363
KKDBP	0.8120	0.9780	0.6450	0.6610
MKSVM (with MKL-CKA)	0.8370	0.9360	0.7420	0.6910
MK-FSVM-SVDD	0.8550	0.9570	0.7530	0.7250
FTWSVM-SR	0.8660	0.9460	0.7850	0.7410
TWSVM	0.8330	0.9460	0.7200	0.6840
DBP-PSSM	0.8118	—	—	0.657
DNAPred_Prot (proposed)	**0.9140**	**0.9785**	**0.8495**	**0.8349**

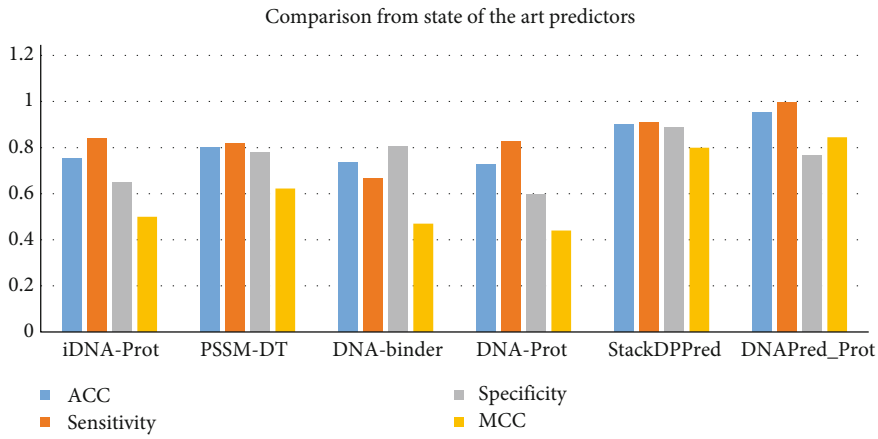


Figure 14: Jackknife results compared with the state-of-the-art predictors.

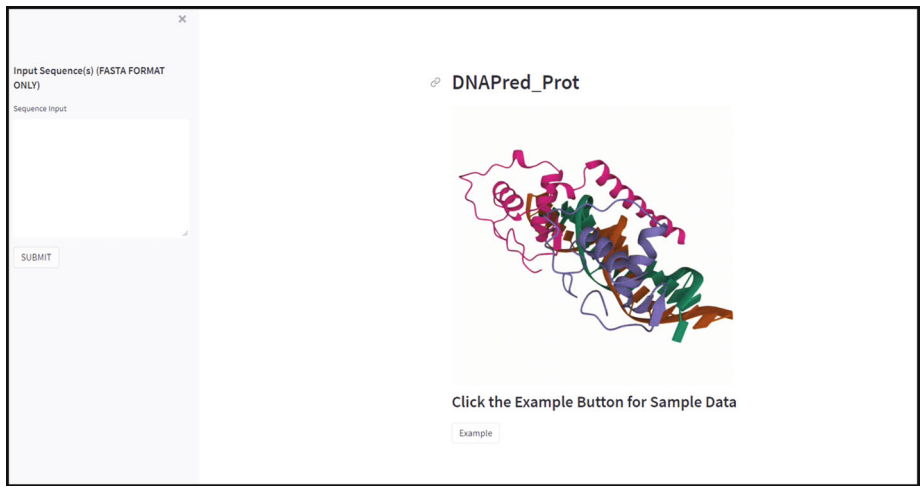


Figure 15: Visualization of the web server.

4.1. 10-Fold Cross-Validation. Using a 10-fold cross-validation testing technique, an accuracy of 94.97%, 48.55%, and 79.5% was achieved with random forest, support vector machine, and artificial neural network, respectively. The results obtained from 10-fold cross-validation via random forest demonstrate that an overall accuracy

obtained is highly acceptable than previously proposed predictors and SVM and ANN classifiers. Overall predicted results obtained from Eq. (23) and comparison with other existing methodologies are shown in Table 4. The ROC comparison for 10-fold, 5-fold cross-validation of random forest, artificial neural network, and support vector machine are shown in Figures 6 and 7, respectively.

4.2. Boxplot Visualization. Box plot is a convenient and straightforward way of displaying a set of data on scale intervals. For analysis of 10-fold cross-validation result boxplots for each classifier RF, ANN, and SVM are shown in Figure 8, Figure 9, and Figure 10, respectively.

4.3. Jackknife Testing. To check the quality of the predictor, we also make use of jackknife testing. In the process of jackknife testing, training and testing datasets are opened, and every sample is lifted between the two. Using this technique "Memory" effect and unforeseen problems can be removed in test and independent dataset subsampling, as from a unique dataset, always the impressive result is obtained by using jackknife testing. Results obtained in the process of recursive training via the random forest are 95.11% accurate, whereas 79.5% and 48.56% accuracy achieved by artificial neural network and support vector machine, respectively, which shows that random forest performs better than the other two classifiers. The results of all three classifiers used in this study are shown in Table 5, while ROC is shown in Figure 11.

4.4. Independent Testing. In independent testing, the dataset is divided into two subsamples, testing and training, first subsample training contains 70% of the dataset and the second testing subsample consists of 30%. Using the random forest technique, 97.33% accurate results were achieved which is better than 20.88% with support vector machine and 79.51% with artificial neural network, training and testing, respectively. The results of all three classifiers used in this study are shown in Table 6, while ROC is shown in Figure 12.

4.5. Self-Consistency. Hastie and Stuetzle in 1989 introduced the term "self-consistency" which becomes the fundamental concept in the field of statistics. It gives the suitable method for a lot of techniques in statistics which led to a more straightforward and more accessible structure for distributions representation by self-consistency, results via random forest obtained are 95.11% accurate, and 79.5% and 48.56% accuracy is obtained by support vector machine and artificial neural network which shows random forest classifier performs better. The results of all three classifiers used in this study are shown in Table 7. Also, the ROC of self-consistency for all three classifiers is shown in Figure 13.

4.6. Comparison with State-of-the-Art Approaches. Using the jackknife testing technique on the standard dataset for the sake of metrics represented in Equation (23), the results obtained by this methodology have an accuracy of 95.11%. To facilitate and comfort, a comparison from the different existing state-of-the-art methodologies with jackknife testing results of this methodology is shown in Table 8 and Table 9. To have a clear view and understanding of the comparison, a bar chart is also shown in Figure 14. It is visible from the table that DNAPred_Prot for metrics, i.e., accuracy, sensitivity, and MCC scores are much high. It indicates that the suggested anticipator is advanced in all four parameters on which the prediction is made for the identification of DNA-binding protein which are stability, sensitivity, specificity, and overall accuracy with its counterparts.

The comparative analysis provided in Table 9 shows that the proposed model with Random-Forest as classifier outperforms all previous existing methods and provides an accuracy of 0.914 on the independent dataset (PDB186).

4.7. Web server. Developing a convenient web server is the 5^{th} step in the five-step rule. As specified and explained in the number of recent publications [73–75, 81, 82], for development of practical, more useful forecasting methods and tools for computation in the future need a web server that is publicly available at the link and easy to use. The user can follow a series of steps to take benefit from the study using a web server. Steps are provided below.

Step 1. Open your browser and go to (https://share.streamlit.io/waqarhusain/dnapred_prot/main/app.py). It can also be seen from Figure 15 that the first page that open is the home page

Step 2. For prediction, input sequence in the sidebar input field. You can also find example data by clicking Example button

Step 3. After entering data, press SUBMIT to perform prediction. Results are shown on the main page in a tabular form. Specifically, a lot of practically important web servers have a rising impact on medical science and get it into a never known before kind of revolution. We serve our attempt for the analysis, examination, and prediction of the approach proposed in this paper by building a web server

5. Conclusion

DNA-binding protein plays a vital role in a lot of biological activities like transcription, DNA recombination, replication, modification, and repair. The present study is dedicated to the identification of DNA-binding protein following the five-step rules. In consideration of this intention, position relative and statistical features were integrated into DNAPred_Prot. Popular verification testing techniques jackknife and cross-validation were utilized to check the proposed model's capability and efficiency. It is crystal clear from the results that random forest performs best among support vector machines and artificial neural networks. Results of a random forest classifier using 10-fold cross-validation and jackknife's approach include 94.97% and 95.11% accurate results achieved, respectively. These results are better as compared to results obtained by support vector machine and artificial neural network. The system's overall accuracy is 95.11% to the sensitivity of 99.75% and specificity of 76.78%. It is to wind up that there is a capability in this model to be more improved in result computation as the number of protein sequences increases.

Data Availability

The data are available through online server: https://share.streamlit.io/waqarhusain/dnapred_prot/main/app.py.

Ethical Approval

It is also declared that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under the grant no. KEP-11-611-39. The authors, therefore, acknowledge DSR for the technical and financial support.

References

[1] M. Ptashne, "Regulation of transcription: from lambda to eukaryotes," *Trends in Biochemical Sciences*, vol. 30, no. 6, pp. 275–279, 2005.

[2] K. A. Jones, J. T. Kadonaga, P. J. Rosenfeld, T. J. Kelly, and R. Tjian, "A cellular DNA-binding protein that activates eukaryotic transcription and DNA replication," *Cell*, vol. 48, no. 1, pp. 79–89, 1987.

[3] M. J. Buck and J. D. Lieb, "ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments," *Genomics*, vol. 83, no. 3, pp. 349–360, 2004.

[4] F. Cajone, M. Salina, and A. Benelli-Zazzera, "4-Hydroxynonenal induces a DNA-binding protein similar to the heat-shock factor," *Biochemical Journal*, vol. 262, no. 3, pp. 977–979, 1989.

[5] H. Zhao, Y. Yang, and Y. Zhou, "Structure-based prediction of DNA-binding proteins by structural alignment and a volume-fraction corrected DFIRE-based energy function," *Bioinformatics*, vol. 26, no. 15, pp. 1857–1863, 2010.

[6] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[7] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 904, no. 1, pp. 23–37, 1995.

[8] H. Tjong and H. X. Zhou, "DISPLAR: an accurate method for predicting DNA-binding sites on protein surfaces," *Nucleic Acids Research*, vol. 35, no. 5, pp. 1465–1477, 2007.

[9] R. E. Langlois and H. Lu, "Boosting the prediction and understanding of DNA-binding domains from sequence," *Nucleic Acids Research*, vol. 38, no. 10, pp. 3149–3158, 2010.

[10] H. L. Huang, I. C. Lin, Y. F. Liou et al., "Predicting and analyzing DNA-binding domains using a systematic approach to identifying a set of informative physicochemical and biochemical properties," *BMC Bioinformatics*, vol. 12, no. S1, p. S47, 2011.

[11] K. Kumar, S. Greenfield, K. Raza, P. Gill, and R. Stack, "Understanding adherence-related beliefs about medicine amongst patients of South Asian origin with diabetes and cardiovascular disease patients: a qualitative synthesis," *BMC Endocrine Disorders*, vol. 16, no. 1, p. 24, 2016.

[12] X. Shao, Y. Tian, L. Wu, Y. Wang, L. Jing, and N. Deng, "Predicting DNA- and RNA-binding proteins from sequences with kernel methods," *Journal of Theoretical Biology*, vol. 258, no. 2, pp. 289–293, 2009.

[13] T. Cui, Y. Dou, P. Tan et al., "RNALocate v2. 0: an updated resource for RNA subcellular localization with increased coverage and annotation," *Nucleic Acids Research*, vol. 50, no. D1, pp. D333–D339, 2022.

[14] Y. D. Cai and A. J. Doig, "Prediction of Saccharomyces cerevisiae protein functional class from functional domain composition," *Bioinformatics*, vol. 20, no. 8, pp. 1292–1300, 2004.

[15] M. Gao and J. Skolnick, "DBD-Hunter: a knowledge-based method for the prediction of DNA–protein interactions," *Nucleic Acids Research*, vol. 36, no. 12, pp. 3978–3992, 2008.

[16] H. P. Shanahan, M. A. Garcia, S. Jones, and J. M. Thornton, "Identifying DNA-binding proteins using structural motifs and the electrostatic potential," *Nucleic Acids Research*, vol. 32, no. 16, pp. 4732–4741, 2004.

[17] N. Bhardwaj, R. E. Langlois, G. Zhao, and H. Lu, "Kernel-based machine learning protocol for predicting DNA-binding proteins," *Nucleic Acids Research*, vol. 33, no. 20, pp. 6486–6493, 2005.

[18] J. B. Brown and T. Akutsu, "Identification of novel DNA repair proteins via primary sequence, secondary structure, and homology," *BMC Bioinformatics*, vol. 10, no. 1, p. 25, 2009.

[19] Y. Xiong, J. Liu, and D. Q. Wei, "An accurate feature-based method for identifying DNA-binding residues on protein surfaces," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 2, pp. 509–517, 2011.

[20] G. Nimrod, M. Schushan, A. Szilágyi, C. Leslie, and N. Ben-Tal, "iDBPs: a web server for the identification of DNA binding proteins," *Bioinformatics*, vol. 26, no. 5, pp. 692-693, 2010.

[21] S. Ahmad and A. Sarai, "Moment-based prediction of DNA-binding proteins," *Journal of Molecular Biology*, vol. 341, no. 1, pp. 65–71, 2004.

[22] M. Andrabi, K. Mizuguchi, A. Sarai, and S. Ahmad, "Prediction of mono- and di-nucleotide-specific DNA-binding sites in proteins using neural networks," *BMC Structural Biology*, vol. 9, no. 1, p. 30, 2009.

[23] X. J. Zhu, C. Q. Feng, H. Y. Lai, W. Chen, and L. Hao, "Predicting protein structural classes for low-similarity sequences by evaluating different features," *Knowledge-Based Systems*, vol. 163, pp. 787–793, 2019.

[24] E. W. Stawiski, L. M. Gregoret, and Y. Mandel-Gutfreund, "Annotating nucleic acid-binding function based on protein structure," *Journal of Molecular Biology*, vol. 326, no. 4, pp. 1065–1079, 2003.

[25] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identification of human microRNAs by incorporating a high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 11, no. 1, pp. 192–201, 2014.

[26] Z. Qian, Y. D. Cai, and Y. Li, "A novel computational method to predict transcription factor DNA binding preference," *Biochemical and Biophysical Research Communications*, vol. 348, no. 3, pp. 1034–1037, 2006.

[27] C. Yan, M. Terribilini, F. Wu, R. L. Jernigan, D. Dobbs, and V. Honavar, "Predicting DNA-binding sites of proteins from amino acid sequence," *BMC Bioinformatics*, vol. 7, no. 1, p. 262, 2006.

[28] G. Govindan and A. S. Nair, "New feature vector for apoptosis protein subcellular localization prediction," in *In International Conference on Advances in Computing and Communications*, Communications in Computer and Information Science, pp. 294–301, Springer, Berlin, Heidelberg, 2011.

[29] L. Nanni and A. Lumini, "Combing ontologies and dipeptide composition for predicting DNA-binding proteins," *Amino Acids*, vol. 34, no. 4, pp. 635–641, 2008.

[30] L. Zhang, Y. Yang, L. Chai et al., "A deep learning model to identify gene expression level using cobinding transcription factor signals," *Briefings in Bioinformatics*, vol. 23, no. 1, p. - bbab501, 2022.

[31] J. F. Xia, X. M. Zhao, and D. S. Huang, "Predicting protein–protein interactions from protein sequences using meta predictor," *Amino Acids*, vol. 39, no. 5, pp. 1595–1599, 2010.

[32] Q. Zou, X. Li, Y. Jiang, Y. Zhao, and G. Wang, "BinMemPredict: a web server and software for predicting membrane protein types," *Current Proteomics*, vol. 10, no. 1, pp. 2–9, 2013.

[33] S. Iqbal and M. T. Hoque, "DisPredict: a predictor of disordered protein using optimized RBF kernel," *PLoS One*, vol. 10, no. 10, article e0141551, 2015.

[34] B. Liu, X. Wang, L. Lin, Q. Dong, and X. Wang, "A discriminative method for protein remote homology detection and fold recognition combining top-n-grams and latent semantic analysis," *BMC Bioinformatics*, vol. 9, no. 1, p. 510, 2008.

[35] B. Liu, X. Wang, Q. Chen, Q. Dong, and X. Lan, "Using amino acid physicochemical distance transformation for fast protein remote homology detection," *PLoS One*, vol. 7, no. 9, article e46633, 2012.

[36] Z. P. Feng and C. T. Zhang, "Prediction of membrane protein types based on the hydrophobic index of amino acids," *Journal of Protein Chemistry*, vol. 19, no. 4, pp. 269–275, 2000.

[37] E. Moroni, M. Caselle, and F. Fogolari, "Identification of DNA-binding protein target sequences by physical effective energy functions: free energy analysis of lambda repressor-DNA complexes," *BMC Structural Biology*, vol. 7, no. 1, p. 61, 2007.

[38] R. Sharma, G. Raicar, T. Tsunoda, A. Patil, and A. Sharma, "OPAL: prediction of MoRF regions in intrinsically disordered protein sequences," *Bioinformatics*, vol. 34, no. 11, pp. 1850–1858, 2018.

[39] B. Wang, P. Chen, D. S. Huang, J. J. Li, T. M. Lok, and M. R. Lyu, "Predicting protein interaction sites from residue spatial sequence profile and evolution rate," *FEBS Letters*, vol. 580, no. 2, pp. 380–384, 2006.

[40] Q. Zou, X. B. Li, W. R. Jiang, Z. Y. Lin, G. L. Li, and K. Chen, "Survey of MapReduce frame operation in bioinformatics," *Briefings in Bioinformatics*, vol. 15, no. 4, pp. 637–647, 2014.

[41] R. Xu, J. Zhou, H. Wang, Y. He, X. Wang, and B. Liu, "Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation," *In BMC Systems Biology*, vol. 9, no. 1, p. S10, 2015.

[42] W. Lou, X. Wang, F. Chen, Y. Chen, B. Jiang, and H. Zhang, "Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naïve Bayes," *PLoS One*, vol. 9, no. 1, article e86703, 2014.

[43] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[44] L. Zhang, X. Zhao, and L. Kong, "Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 355, pp. 105–110, 2014.

[45] Y. Zhang, P. Chen, Y. Gao, J. Ni, and X. Wang, "DBP-PSSM: combination of evolutionary profiles with the XGBoost algorithm to improve the identification of DNA-binding proteins," *Combinatorial Chemistry & High Throughput Screening*, vol. 25, no. 1, pp. 3–12, 2022.

[46] K. Harini, A. Srivastava, A. Kulandaisamy, and M. M. Gromiha, "ProNAB: database for binding affinities of protein–nucleic acid complexes and their mutants," *Nucleic Acids Research*, vol. 50, no. D1, pp. D1528–D1534, 2022.

[47] Y. Jia, S. Huang, and T. Zhang, "KK-DBP: A multi-feature fusion method for DNA-binding protein identification based on random forest," *Frontiers in Genetics*, vol. 12, p. 2458, 2021.

[48] J. Hu, L. Rao, Y. H. Zhu, G. J. Zhang, and D. J. Yu, "TargetDBP +: enhancing the performance of identifying DNA-binding proteins via weighted convolutional features," *Journal of Chemical Information and Modeling*, vol. 61, no. 1, pp. 505–515, 2021.

[49] Y. Qian, L. Jiang, Y. Ding, J. Tang, and F. Guo, "A sequence-based multiple kernel model for identifying DNA-binding proteins," *BMC Bioinformatics*, vol. 22, no. S3, pp. 1–18, 2021.

[50] Y. Zou, Y. Ding, L. Peng, and Q. Zou, "FTWSVM-SR: DNA-binding proteins identification via fuzzy twin support vector machines on self-representation," in *Interdisciplinary Sciences: Computational Life Sciences*, 2021.

[51] Y. Zou, H. Wu, X. Guo et al., "MK-FSVM-SVDD: a multiple kernel-based fuzzy SVM model for predicting DNA-binding proteins via support vector data description," *Current Bioinformatics*, vol. 16, no. 2, pp. 274–283, 2021.

[52] C. Zou, J. Gong, and H. Li, "An improved sequence based prediction protocol for DNA-binding proteins using SVM and comprehensive feature analysis," *BMC Bioinformatics*, vol. 14, no. 1, p. 90, 2013.

[53] Z. Zhang, S. Kochhar, and M. G. Grigorov, "Descriptor-based protein remote homology identification," *Protein Science*, vol. 14, no. 2, pp. 431–444, 2005.

[54] M. Arif, M. Hayat, and Z. Jan, "iMem-2LSAAC: a two-level model for discrimination of membrane proteins and their types by extending the notion of SAAC into Chou's pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 442, pp. 11–21, 2018.

[55] J. Mei and J. Zhao, "Prediction of HIV-1 and HIV-2 proteins by using Chou's pseudo amino acid compositions and different classifiers," *Scientific Reports*, vol. 8, no. 1, p. 2359, 2018.

[56] J. Mei and J. Zhao, "Analysis and prediction of presynaptic and postsynaptic neurotoxins by Chou's general pseudo amino acid composition and motif features," *Journal of Theoretical Biology*, vol. 447, pp. 147–153, 2018.

[57] A. Ashraf, A. Shahzadi, and M. S. Akram, "Protein carbonylation sites prediction using biomarkers of oxidative stress in various human diseases: a systematic literature review," *Vfast transaction on Software Engineering*, vol. 9, pp. 20–29, 2021.

[58] L. Zhang and L. Kong, "iRSpot-ADPM: identify recombination spots by incorporating the associated dinucleotide product model into Chou's pseudo components," *Journal of Theoretical Biology*, vol. 441, pp. 1–8, 2018.

[59] S. Zhang and X. Duan, "Prediction of protein subcellular localization with oversampling approach and Chou's general PseAAC," *Journal of Theoretical Biology*, vol. 437, pp. 239–250, 2018.

[60] N. Albugami, "Prediction of Saudi Arabia SARS-COV 2 diversifications in protein strain against China strain," *VAWKUM Transactions on Computer Sciences*, vol. 8-1, pp. 63–74, 2020.

[61] H. Lv, Y. Zhang, J. S. Wang et al., "iRice-MS: an integrated XGBoost model for detecting multitype post-translational modification sites in rice," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab486, 2022.

[62] S. J. Malebary and Y. D. Khan, "Identification of antimicrobial peptides using Chou's 5 step rule," *CMC-Computers Materials & Continua*, vol. 67, no. 3, pp. 2863–2881, 2021.

[63] O. Barukab, F. Ali, and S. A. Khan, "DBP-GAPred: an intelligent method for prediction of DNA-binding proteins types by enhanced evolutionary profile features with ensemble learning," *Journal of Bioinformatics and Computational Biology*, vol. 19, no. 4, p. 2150018, 2021.

[64] R. C. Papademetriou, "Reconstructing with moments," *In Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. III. Conference C: Image, Speech and Signal Analysis*, pp. 476–480, IEEE Computer Society Press, 1992.

[65] S. J. Malebary and Y. D. Khan, "Evaluating machine learning methodologies for identification of cancer driver genes," *Scientific Reports*, vol. 11, no. 1, pp. 1–13, 2021.

[66] J. Chen, H. Liu, J. Yang, and K. C. Chou, "Prediction of linear B-cell epitopes using amino acid pair antigenicity scale," *Amino Acids*, vol. 33, no. 3, pp. 423–428, 2007.

[67] K. C. Chou, "Using subsite coupling to predict signal peptides," *Protein Engineering*, vol. 14, no. 2, pp. 75–79, 2001.

[68] W. R. Qiu, B. Q. Sun, X. Xiao, Z. C. Xu, and K. C. Chou, "iPTM-mLys: identifying multiple lysine PTM sites and their different types," *Bioinformatics*, vol. 32, no. 20, pp. 3116–3123, 2016.

[69] S. Murad, A. Mashat, A. Mahfooz, S. A. Khan, and O. Barukab, *UbiSites-SRF: Ubiquitination Sites Prediction Using Statistical Moment with Random Forest Approach*, Research Square, 2021.

[70] H. Zulfiqar, Z. J. Sun, Q. L. Huang et al., "Deep-4mCW2V: A sequence-based predictor to identify N4-methylcytosine sites in *Escherichia coli*," *Methods*, 2021.

[71] H. Lv, F. Y. Dao, D. Zhang, H. Yang, and H. Lin, "Advances in mapping the epigenetic modifications of 5-methylcytosine (5mC), N6-methyladenine (6mA), and N4-methylcytosine (4mC)," *Biotechnology and Bioengineering*, vol. 118, no. 11, pp. 4204–4216, 2021.

[72] H. Lv, F. Y. Dao, H. Zulfiqar, and H. Lin, "DeepIPs: comprehensive assessment and computational identification of phosphorylation sites of SARS-CoV-2 infection using a deep learning-based approach," *Briefings in Bioinformatics*, vol. 22, no. 6, p. 244, 2021.

[73] A. H. Butt and Y. D. Khan, "CanLect-Pred: a cancer therapeutics tool for prediction of target cancerlectins using experiential annotated proteomic sequences," *IEEE Access*, vol. 8, pp. 9520–9531, 2019.

[74] S. Naseer, W. Hussain, Y. D. Khan, and N. Rasool, "Sequence-based identification of arginine amidation sites in proteins using deep representations of proteins and PseAAC," in *Current Bioinformatics*, Bentham Science Publishers, 2021.

[75] Z. Y. Zhang, Z. J. Sun, Y. H. Yang, and H. Lin, "Towards a better prediction of subcellular location of long non-coding RNA," *Frontiers of Computer Science*, vol. 16, no. 5, pp. 1–7, 2022.

[76] A. Tyryshkina, N. Coraor, and A. Nekrutenko, "Predicting runtimes of bioinformatics tools based on historical data: five years of galaxy usage," *Bioinformatics*, vol. 35, no. 18, pp. 3453–3460, 2019.

[77] N. Simidjievski, L. Todorovski, and S. Džeroski, "Modeling dynamic systems with efficient ensembles of process-based models," *PLoS One*, vol. 11, no. 4, p. e0153507, 2016.

[78] R. E. Schapire, "Theoretical, views of boosting and applications," in *In International Conference on Algorithmic Learning Theory*Springer, Berlin, Heidelberg.

[79] D. Wang, Z. Zhang, Y. Jiang et al., "DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism," *Nucleic Acids Research*, vol. 49, no. 8, pp. e46–e46, 2021.

[80] F. Y. Dao, H. Lv, H. Zulfiqar et al., "A computational platform to identify origins of replication sites in eukaryotes," *Briefings in Bioinformatics*, vol. 22, no. 2, pp. 1940–1950, 2021.

[81] M. Shahid, M. Ilyas, W. Hussain, and Y. D. Khan, "ORI-deep: improving the accuracy for predicting origin of replication sites by using a blend of features and long short-term memory network," *Briefings in Bioinformatics*, 2022.

[82] S. Amanat, A. Ashraf, W. Hussain, N. Rasool, and Y. D. Khan, "Identification of lysine carboxylation sites in proteins by integrating statistical moments and position relative features via general PseAAC," *Current Bioinformatics*, vol. 15, no. 5, pp. 396–407, 2020.