Prediction of Recombination Spots Using Novel Hybrid Feature Extraction Method via Deep Learning Approach

Fatima Khan[1], *Mukhtaj Khan*[1]*, *Nadeem Iqbal*[1]*, *Salman Khan*[1], *Dost Muhammad Khan*[2], *Abbas Khan*[3] and *Dong-Qing Wei*[3,4,5]

[1] Department of Computer Science, Abdul Wali Khan University Mardan, Mardan, Pakistan, [2] Department of Statistics, Abdul Wali Khan University Mardan, Mardan, Pakistan, [3] Department of Bioinformatics and Biological Statistics, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, China, [4] State Key Laboratory of Microbial Metabolism, Shanghai-Islamabad-Belgrade Joint Innovation Center on Antibacterial Resistances, Joint Laboratory of International Cooperation in Metabolic and Developmental Sciences, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Ministry of Education, Shanghai, China, [5] Peng Cheng Laboratory, Shenzhen, China

OPEN ACCESS

Edited by:
Xianwen Ren,
Peking University, China

Reviewed by:
Yongcui Wang,
Northwest Institute of Plateau Biology (CAS), China
Tahir Hameed,
Merrimack College, United States

***Correspondence:**
Mukhtaj Khan
mukhtaj.khan@awkum.edu.pk
Nadeem Iqbal
nikhan@awkum.edu.pk

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: 10 March 2020
Accepted: 13 August 2020
Published: 17 September 2020

Citation:
Khan F, Khan M, Iqbal N, Khan S, Muhammad Khan D, Khan A and Wei D-Q (2020) Prediction of Recombination Spots Using Novel Hybrid Feature Extraction Method via Deep Learning Approach. Front. Genet. 11:539227. doi: 10.3389/fgene.2020.539227

Meiotic recombination is the driving force of evolutionary development and an important source of genetic variation. The meiotic recombination does not take place randomly in a chromosome but occurs in some regions of the chromosome. A region in chromosomes with higher rate of meiotic recombination events are considered as hotspots and a region where frequencies of the recombination events are lower are called coldspots. Prediction of meiotic recombination spots provides useful information about the basic functionality of inheritance and genome diversity. This study proposes an intelligent computational predictor called iRSpots-DNN for the identification of recombination spots. The proposed predictor is based on a novel feature extraction method and an optimized deep neural network (DNN). The DNN was employed as a classification engine whereas, the novel features extraction method was developed to extract meaningful features for the identification of hotspots and coldspots across the yeast genome. Unlike previous algorithms, the proposed feature extraction avoids bias among different selected features and preserved the sequence discriminant properties along with the sequence-structure information simultaneously. This study also considered other effective classifiers named support vector machine (SVM), K-nearest neighbor (KNN), and random forest (RF) to predict recombination spots. Experimental results on a benchmark dataset with 10-fold cross-validation showed that iRSpots-DNN achieved the highest accuracy, i.e., 95.81%. Additionally, the performance of the proposed iRSpots-DNN is significantly better than the existing predictors on a benchmark dataset. The relevant benchmark dataset and source code are freely available at: https://github.com/Fatima-Khan12/iRspot_DNN/tree/master/iRspot_DNN.

Keywords: DNA sequence, feature selection, deep neural network, classification, system biology, novel feature extraction

INTRODUCTION

Meiotic recombination is the process of exchanging alleles between homologous chromosomes, which take place during meiosis (Lichten and Goldman, 1995; Petes, 2001; Liu et al., 2012, 2016). It is a vital biological process that is carried out in two phases; meiosis and recombination. In meiosis, the genome is divided into two equivalent parts which are known as daughter cells that take part in the production of a new living organism. While in the recombination process, the different gametes are joined to make new genetics combinations (Kabir and Hayat, 2016). It is essential for cell division and an important process to make heredity variances (Paul et al., 2016; Zhang and Kong, 2018b). Hence, the meiotic recombination gives opportunities for natural exchange of hereditary variations (Chen et al., 2013; Zhang and Kong, 2018b), which causes the genome to create more hereditary differences and speed up the genetic progress. The meiotic recombination does not take place randomly in a chromosome but occurs in some regions of a chromosome. In general, the region that exhibits a high frequency of recombination is considered as hotspots, whereas the region that exhibits low frequency of recombination is considered as coldspots (Liu et al., 2012; Dong et al., 2016). The study of recombination spots provides useful information about the basic functionality of inheritance and genome diversity. Additionally, it gives valuable insights about variation in DNA sequence and patterns, which may help to know the position of alleles that cause different diseases (Abeysinghe et al., 2003; Hey, 2004).

Owning to the importance of meiotic recombination, several predictors have been introduced in the literature using machine learning methods for identification of recombination spots (Zhou et al., 2006; Jiang et al., 2007b; Liu et al., 2012, 2016, 2017a; Chen et al., 2013; Li et al., 2014; Qiu et al., 2014; Dong et al., 2016; Kabir and Hayat, 2016; Wang et al., 2016; Dwivedi, 2018). For example, Liu et al. (2012) proposed a model for discrimination of recombination spots using an increment of diversity with quadratic discriminant analysis (IDQD) method and k-mer approach. Jiang et al. (2007b) proposed RF-DYMH predictor based on gapped dinucleotide composition (GDC) technique for feature formulation and RF as a classification algorithm. Chen et al. (2013) proposed iRSpot-PseDNC based on "pseudo dinucleotide composition" (PseDNC) with SVM. The authors employed PseDNC with physiochemical properties for feature extraction and SVM as a classification engine. Liu et al. (2016) proposed iRSpot-DACC based on dinucleotide-based autocross covariance (DACC) with SVM as a learning algorithm. The DACC incorporated global sequence order information along with local DNA properties to construct a feature vector. Liu et al. (2017a) proposed iRSpot-EL model for discrimination of recombination spots. The proposed model applied PseKNC and DACC along with ensemble classifier. Kabir and Hayat (2016) proposed iRSpot-GAEnsC using different sequence formulation methods, such as nucleotide, di-nucleotide, and tri-nucleotide along with ensemble classifiers. Qiu et al. (2014) proposed iRSpot-TNCPseAAC that combined TNC and PseAAC (pseudo amino acid composition) techniques to formulate DNA samples. The TNC method was used to integrate DNA local or short-range sequence order information, whereas, the PseAAC method was applied to integrate DNA global and long-range sequence order information. Zhang and Kong (2018b) proposed iRSpot-PDI using PseAAC as a feature extraction technique along with the BFS (best first search) method for feature selection. Maruf and Shatabda (2018) proposed iRSpot-SF computational model for the identification of hotspot using a sequence based feature method with SVM. The author used different K-mer composition approaches to extract optimum features. Zhang and Kong (2018a) proposed iRSpot-ADPM using di-nucleotide composition (DNC) as a sequence formulation technique and SVM as a classification engine. Jani et al. (2018) proposed iRecSpot-EF for the classification of recombination hotspots and coldspots. The authors employed K-mer for feature extraction, AdaBoost technique for feature selection, and logistic regression as classification algorithms. The methods mentioned above have applied single layer conventional machine learning methods that are unable to discriminate hotspots and coldspots accurately.

Recently, Khan et al. (2019b) proposed iRSpot-SPI for predictions of hotspots and coldspots based on multilayer deep learning algorithm. The proposed model used sequence-structure information along with deep learning as a discriminative method. The proposed model achieved the highest accuracy; however, the authors ignored optimization (i.e., tuning) of hyper-parameters of the DNN model. We argue that with un-tuning parameters, a model can generate unstable results, which affect the overall performance of the model. In short, the existing models are summarized in **Table 1** according to applied feature extraction methods and machine learning algorithms.

This paper proposes an intelligent computation model based on a novel hybrid feature extraction method along with optimized DNN for the prediction of recombination hotspots and coldspots. Moreover, the proposed model employed SVM-RFE (Guyon et al., 2002; Zhang et al., 2006) for discriminant feature selection. The proposed model was designed to follow Chou's five-steps rule mentioned comprehensively in a series of publications (Chou et al., 2011; He et al., 2015; Jia et al., 2015; Khan et al., 2019a). The framework of the proposed iRSpot-DNN is shown in **Figure 1**. Firstly, we selected a valid benchmark dataset that contained recombination hotspots and coldspots sequences. The benchmark dataset was split into training and testing dataset. Secondly, different feature extraction methods were employed to construct feature vectors. Thirdly, we obtained discriminant features using feature selection method. Fourthly, we proposed a novel method that considered the contribution of different feature extraction methods in order to avoid biasness and preserved sequence discriminative properties. Fifthly, the proposed model applied a grid search approach for hyper-parameters tuning. Sixthly, the DNN model with optimized hyper-parameters was applied to predict recombination spots. Finally, the performance of the proposed model was evaluated on a selected benchmark dataset using a 10-fold cross-validation test. Based on evaluation results, the iRSpot-DNN yielded the highest success rate of 95.81%, sensitivity of 96.17%, specificity of 95.92%, and Matthews correlation coefficient of 0.915. Furthermore, the outcome of iRSpot-DNN was compared with the current

TABLE 1 | Summery of exiting model according to feature extraction methods and machine learning algorithms.

Model name	Classification algorithm	Feature extraction method
RF-DYMHC (Jiang et al., 2007b)	RF	GDC
IDQD (Liu et al., 2012)	IDQD	K-mer
iRSpot-PseDNC (Chen et al., 2013)	SVM	PseDNC
iRSpot-TNCPseAAC (Qiu et al., 2014)	SVM	TNC and PseAAC
iRSpot-GAEnsc (Kabir and Hayat, 2016)	KNN, SVM, RF	DNC and TNC
iRSpot-DACC (Liu et al., 2016)	SVM	DACC
iRSpot-EL (Liu et al., 2017a)	SVM	DACC and PseKNC
iRSpot-ADPM (Zhang and Kong, 2018a)	SVM	DNC
iRSpot-SF (He et al., 2018)	SVM	K-mer Composition, TF-IDF, gapped k-mer composition, and reverse complement k-mer composition (RCC)
iRecSpot-EF (Jani et al., 2018)	Logistic regression	RCC
iRSpot-SPI (Khan et al., 2019b)	DNN	GDC, RCC, and PseTNC

predictors, and the comparison results illustrated that the proposed iRSpot-DNN outperformed the current predictors.

METHOD AND MATERIAL

Benchmark Dataset

The selection of a consistent and standard benchmark dataset is the first step toward building an intelligent, accurate, and reliable prediction model. In order to build a reliable prediction model, this paper selected a standard benchmark dataset presented in Jiang et al. (2007b), which is used in several papers (Liu et al., 2012; Chen et al., 2013; Qiu et al., 2014; Yang et al., 2018; Zhang and Kong, 2018b; Khan et al., 2019b). We formulated the benchmark dataset as follows:

$$S = S^{+} \cup S^{-} \tag{1}$$

Here S^+ represents hotspots and S^- represent coldspots. *U* is the set theory operator representing a union of both the S^+ and S^-. Initially, the dataset contained 490 S^+ sequences and 591 S^- sequences. To eliminate redundant and homologous sequences, we applied CD-HIT (Li and Godzik, 2006; Fu et al., 2012) software that removed sequences having similarity more than 75%. The updated dataset contained 478 S^+ sequences and 572 S^- sequences.

Sequence Formulation Methods

In the previous section, we discussed the construction of a benchmark dataset that contained DNA hotspots and coldspots sequences. In this section, we formulate biological sequences of different lengths in a feature vector with the same length. Since the statistical machine learning models deal only with numerical descriptors of equal length (Chou, 2015; Noi and Kappas, 2018). Therefore, the biological sequences are required to transform (formulate) into a uniform discrete feature vector before they are given to a computation model. However, the biological sequence may lose pattern or order information at the time of the sequence formulation process. Therefore, various methods in the area of computational biology have been proposed for formulation of DNA, RNA and protein sequences into a distinct feature vector with preserved the sequence pattern and order information (Chen et al., 2013; Qiu et al., 2014; Kabir and Hayat, 2016; Liu et al., 2016; Wang et al., 2016; Yang et al., 2018; Zhang and Kong, 2018b). Besides, web servers have been developed that can be used to convert DNA, RNA, and protein sequences into features vectors according to user's need (Liu et al., 2015, 2017b; Chen et al., 2019).

In this paper, firstly, we applied different sequence formulation/feature extraction methods, such as Gapped di-nucleotide composition (GDC), Reverse complement composition (RCC) and PseTri-Nucleotide Composition (PseTNC) to transform biological sequences into feature vectors. Secondly, we derive a novel formula that generated different features groups based on contribution of the selected feature.

Let suppose D is a DNA sequence from dataset S with length L, can be expressed in mathematical form as Equation (2).

$$\mathrm{D} = \mathrm{D}_1, \mathrm{D}_2, \mathrm{D}_3, \ldots \mathrm{D}_\mathrm{L} \tag{2}$$

Where $\mathrm{D}_1 \in \{A, C, G, T\}$ and $\mathrm{i} \in (1, 2, 3 \ldots \mathrm{L})$. D_1 is the nucleotide at first residue position; D_2 is the nucleotide at second residue position and so no.

Gapped Di-nucleotide Composition

Di-nucleotide composition method is widely used for a sequence formulation; however, this method is considered a correlation between two nucleotides having the same properties. In order to consider K intervals correlation, we used GDC that describes the correlation of every two pairs of nucleotides with a total number of K intervals in a sequence. Many research papers have been applied to the GDC (Jiang et al., 2007b; Ghandi et al., 2014; Tang et al., 2016; Maruf and Shatabda, 2018; Khan et al., 2019b) as a feature extraction method. The GDC computes the cumulative frequency of every two pairs of nucleotides with k number of intervals in a sequence. The GDC can be formulated using Equation (3).

$$g^{\mathrm{i}}_{(\kappa)} = \frac{O^{\mathrm{i}}_{(\kappa)}}{n_{(\kappa)}} \tag{3}$$

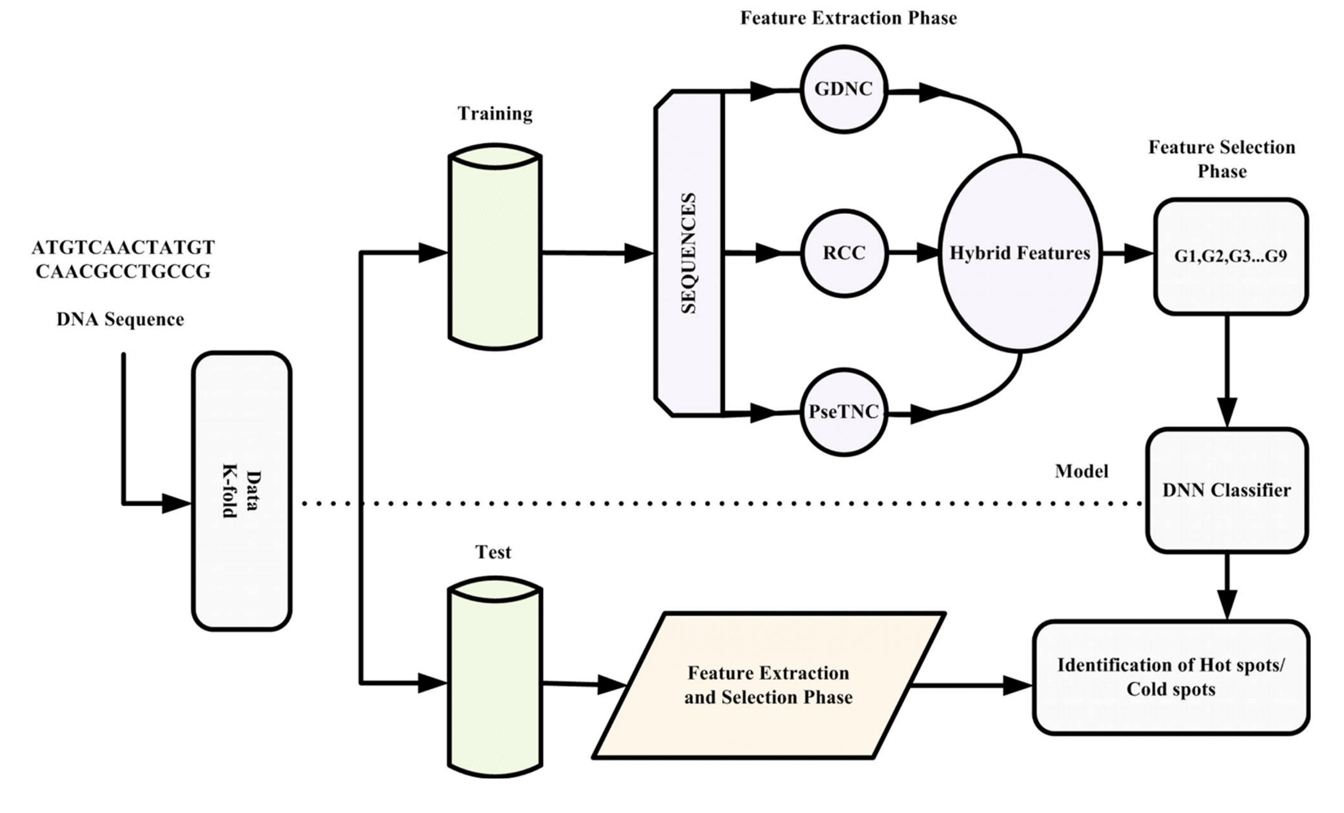

FIGURE 1 | Framework of proposed iRSpot-DNN.

Where $O^{\mathrm{i}}_{(\kappa)}$ is the number of i^{th} observed in two nucleotides, k is the intervals of bases and $n_{(\kappa)}$ is the entire population size of two nucleotide with k interval bases (Tang et al., 2016).

Reverse Complement Composition

The reverse complement of a sequence can be achieved by reversing the letters of a sequence, i.e., exchanging A and T and exchanging C and G. A genome sequence obscures valuable information in a hidden pattern as well as in a reverse complement pattern, that provides most important regularity information (Lopez et al., 1999). Rev(k-mer) composition can be expressed as Equation (4):

$$RC_{\mathrm{Composition}}(S_{\mathrm{i}}) = \frac{1}{L} Count_{\mathrm{R}}(S_{\mathrm{i}}), \forall k = 3, 4, 5 \ldots \tag{4}$$

PseTri-Nucleotide Composition

The PseTri-Nucleotide composition (PseTNC) method was introduced by Chou's et al. for a sequence formulation. The PseTNC method considers three nucleotide compositions (NC) during the sequence formulation and also considers preserving sequence order information (Kabir and Yu, 2017; Khan et al., 2020). In PseTNC, the occurrence frequency of three NC can be computed using the method mentioned in Du et al. (2012) and Li et al. (2016). The PseTNC method can be represented in general form with K-tuple as Equation (5).

$$\mathrm{D} = \left[f_1^{K-tuple} f_2^{K-tuple} \ldots f_i^{K-tuple} \ldots f_{4^k}^{K-tuple} \right]^T \tag{5}$$

$f_i^{K-tupple}$ is the normalized frequency of ith k-tuple nucleotide in a DNA sequence. We can observe from Equation (5) that increasing the value of K, increases the dimensionality of the feature vector. In order to limit the feature vector dimension to 64 possible combinations, we re-write Equation (5) in the form of 3-tuple PseTNC as Equation (6):

$$\mathrm{D} = \left[f_1^{3-tuple} f_2^{3-tuple} \ldots f_{64}^{3-tuple} \right]^T \tag{6}$$

Where, $f_1^{3-tuple} = f(AAA), f_2^{3-tuple} = f(AAC), \ldots f_{64}^{3-tuple} = f(TTT)$

Equation (6) can be written in terms of Equation (2)

$$\mathrm{D} = [D_1 \; D_2 \ldots D_{64} \; D_{64+1} \ldots D_{64+\lambda}]^T \tag{7}$$

$$d_{\mathrm{v}} = \begin{cases} \dfrac{f_v^{3-tuple}}{\sum_{i=1}^{64} f_i^{3-tuple} + w \sum_{j=1}^{\lambda} \theta_j} & 1 \le v \le 64 \\ \dfrac{w\theta_{v-64}}{\sum_{i=1}^{64} f_i^{3-tuple} + w \sum_{j=1}^{\lambda} \theta_j} & (64+1) \le v \le (64+\lambda) \end{cases} \tag{8}$$

TABLE 2 | Number of features generated by different feature extraction method.

Feature extraction method	Number of features
Gapped dinucleotide composition	128
Reverse complement composition	680
PseTNC	66

TABLE 3 | Summary of selected features.

Feature extraction method	Number of features	Selected features
Gapped dinucleotide composition	128	5
Reverse complement composition	680	12
PseTNC	66	66

TABLE 4 | Number of features in each group computed using Equation (10).

Feature group	A	λ	Total features
G1	0	0	425
G2	0	1	78
G3	1	0	430
G4	1	1	83
G5	0.5	0	428
G6	0	0.5	252
G7	0.5	0.5	255
G8	0.5	1	81
G9	1	0.5	257

In Equation (8) w denotes weight factor and θ denotes correlation factor, given as follows:

$$\theta_j = \frac{1}{L^* - j}\Sigma_{i=1}^{L^*-j}\theta\left(T_i;\, T_{i+j}\right) j = 1, 2, \ldots \lambda < L^* \tag{9}$$

Discriminant Feature Selection

In the previous section we described different feature extraction/formulation methods that generate various numbers of features as shown in **Table 2**. Feature vector play a vital role in a model prediction process, however, a feature vector with a high dimension space may negatively effects the outcome of a prediction model due to noisy, redundant, and irrelevant features. A number of techniques have been proposed in the literature to reduce feature vector dimensionality by removing the redundant, noisy and irrelevant features. In this paper we employed SVM-RFE (SVM-Recursive Feature Elimination) (Guyon et al., 2002; Zhang et al., 2006) technique as a feature selection method to reduce dimensionality of a feature vector with minimum loss of discriminative features. As a result we obtained selected features vectors summarized in **Table 3**. It is to be noted that we employed SVM-RFE method on PseTNC feature vector to eliminate noisy feature and to obtain selected features, however, the selected features could not significantly improved the performance of the model. Hence, we utilized all the generated features, i.e., 66 of the PseTNC in prediction of recombination spots.

Hybrid Feature and Feature Selection

In this section, we derived a novel formula based on feature extract methods, such as GDC, RCC, PseTNC, and feature extracted in iRecSpot-EF (Jani et al., 2018) and their contributions. The novel formula can be written as Equation (10).

$$\mathcal{F} = a\mathrm{G} + \lambda\,(\,\mathrm{R} + \mathrm{PseTNC}) + (1 - \lambda)(H) \tag{10}$$

In Equation (10) "a" and "λ" are two parameters having values between (0,1), which represent the contribution of a method in the feature vector. G represents GDC, R represents RCC, PseTNC represents PseTri-Nucleotide Composition and H represents 425 features generated by iRecSpot-EF (Jani et al., 2018). Further, G represents five selected features, R represents 12 selected features and PseTNC represents 66 features (Khan et al., 2019b). The parameters "a" and "λ" are used to measure the contribution of the feature extraction methods in the dimension of a hybrid feature vector. A different combination of "a" and "λ" generates various numbers of features, as shown in **Table 4**. For example, using a combination of 0 (i.e., "a" = 0 and "λ" = 0), a total of 425 features are generated, which is represented by G1. Similarly, a combination of 0 and 1 (i.e., "a" = 0 and "λ" = 1) generates 78 features, which is represented by G2 and so on. Hence, the number of selected features are based on the values of "a" and "λ."

Classification Algorithms

Deep Neural Network

Deep learning algorithms apply neural networks that learn features from the data directly and then make decisions. Recently, deep learning algorithms received significant attention in the field of bioinformatics and computational biology (Lecun et al., 2015; Angermueller et al., 2016; Kelley et al., 2016; Mamoshina et al., 2016; Quang and Xie, 2016; Min et al., 2017; Cohn et al., 2018; Miao and Miao, 2018; Telenti et al., 2018; Zhang et al., 2019). A DNN model comprises an input layer, output layer, and multiple hidden layers, as shown in **Figure 2**. The given input data is passed through each layer, where the output of the previous layer is presented as input to the next layer.

The performance (i.e., accuracy) of the DNN model depends upon the number of hidden layers in a network. In general, a network configures with a large number of hidden layers in a training or testing phase can lead to excellent learning and ultimately improve the accuracy of the model (Khan et al., 2019a). However, it may lead to major problems, such as the complexity of the model, computation cost, and overfitting (Liu et al., 2015; Chen et al., 2019).

Deep learning methods have been successfully employed in several areas, including speech recognition (Deng et al., 2012; Sainath et al., 2013), image processing (Krizhevsky et al., 2012; Couprie et al., 2013; Tompson et al., 2014), natural language processing (Mikolov et al., 2011; Bordes and Weston, 2014), bio-Engineering (Acharya et al., 2018; Zhu et al., 2019) and genomics (Khan et al., 2019b). In addition, different research

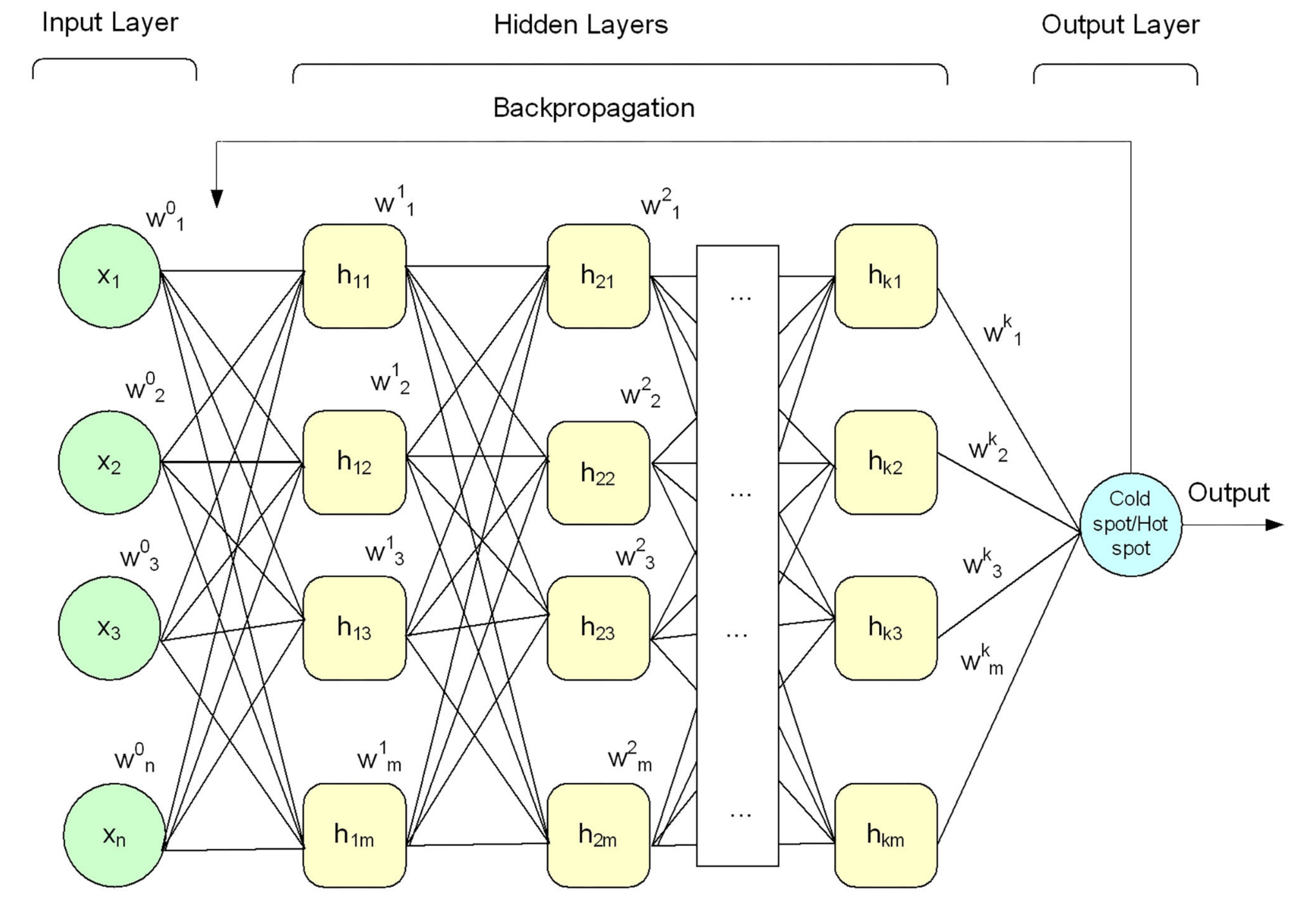

FIGURE 2 | Architecture of deep neural network.

papers have proved that the deep learning methods performed better than conventional machine learning techniques used for various complex learning problems (Deng et al., 2012; Leung et al., 2014; Ma et al., 2015). Due to the remarkable performance of the deep neural network in different domains, in this paper, we apply the DNN model as a classifier for the prediction of recombination spots.

In this paper, the DNN model was configured with a small number of hidden layers (i.e., five hidden layers) along with input layer and output layer as shown in **Figure 2** to keep the model simple (i.e., less computationally costly) and avoid the model overfitting problem. Each layer of the model was configured with multiple processing nodes (i.e., neurons). Firstly, a given feature vector $X\{x_1x_2x_3\ldots x_n\}$ was fed to the input layer and computed output using Equation (11). Secondly, the output of the input layer was fed as input to the first hidden layer and produced a new output. Thirdly, the output of the first hidden layer was provided as input to the second hidden layer and computed output again. This process was continued till we reach to the output layer. The output layer generated binary value, i.e., 0 and 1. The value 0 represents hotspot, and 1 represents coldspots. Furthermore, different activation functions were employed at the input layer and hidden layers, however, the DNN model with hyperbolic tangent (Tanh) activation function generated promising results compared with other activation functions (see **Table 6**). The softmax function was applied at the output layer of the deep neural network to map the output (non-normalized output) of the last layer to a probability distribution to predict the output class. Moreover, stochastic gradient descent and backpropagation were used to optimize weight and bias value to minimize the error. In addition, we employed regularization and dropped out methods to overcome any possible occurrence of the model overfitting issue. Mathematically, a single layer computation can be expressed as Equation (11).

$$Y = g(\sum_{i=1}^{n} Xw_i^k + b_i) \tag{11}$$

Where g represents activation function, X represents feature vector, n represents the number of features, k represents the layers, and b represents bias value.

Support Vector Machine

Support vector machine (SVM) is an effective supervised machine learning technique mostly used for classification and regression. SVM method was first introduced by Cortes and Vapnik (1995) for binary classification problems; however, later on, it was modified for multiclass problems (Ahmad et al., 2015). SVM converts the input data into a high dimensional features space based on transformation, and then define the best possible separating hyper-plane (Qiu et al., 2014). The key points of SVM are the ability to handle large and noisy datasets while avoiding overfitting (Zavaljevski et al., 2002). SVM algorithm can be applied with different kernels; however, SVM with Gaussian Radial Basis Function (RBF) generally generated promising results. Additionally, the SVM can be configured with two other parameters, i.e., C, used for controlling the cost of misclassifications, and γ, used for handling the non-linear classification (Qian et al., 2015; Ballanti et al., 2016). Further details on SVM and its parameter optimization are given in Chou and Elrod (2002) and Cai et al. (2003).

K-nearest Neighbor

K-nearest neighbor (KNN) (or Lazy learning) algorithm is a popular algorithm used for both classification and regression purposes (Hu et al., 2016). However, it is mostly used for classification problems (Ali et al., 2015; Zuo et al., 2015; Khan et al., 2017). KNN is an instance-based and non-parametric learning algorithm and can be considered a simple machine learning algorithm (Donaldson, 1967; Qin et al., 2013). KNN algorithm applies Euclidian distance formula to compute distance amongst the instances for classification. The principal characteristic of the KNN is minimum computation times during the training phase; however, it takes a long time during the testing phase. In KNN algorithm, the value of K plays a significant role and it is used to control the fine-tuning of the algorithm. The model becomes less stable when the value of K decreases, Inversely the model go toward more stability when the value of K increases (Harrison, 2018). The KNN algorithm generates promising performance on the dynamic type of data that changes and updates quickly (Van Der Malsburg, 1986; Kondarasaiah and Ananda, 2004). The KNN algorithm becomes slower when the number of samples or examples increases.

TABLE 5 | List of hyper-parameters with optimized values.

S. No	Parameter	Optimum configuration value
01	Training iterations	1,000
02	Learning rates	0.1
03	Activation function at output layer	Softmax
04	Activation function at hidden layer	Tanh
05	Seed	6
06	Number of hidden layers	4
07	Number of neuron at hidden layers	430-413-318-251-182-96-2
08	Weight initialization function	XAVIER function
09	Optimizer	SGD method
10	Momentum	0.9
11	Updater	ADAGRAD function

TABLE 6 | Impact of learning rates and activation functions on the accuracy of DNN model.

Learning rates	Tanh (%)	ReLU (%)	Sigmoid (%)
0.08	95.43	93.71	54.48
0.09	95.14	93.62	54.48
0.1	95.81	93.52	54.48
0.2	94.86	93.71	89.05
0.3	95.05	93.33	94.86
0.4	93.14	93.81	95.24
0.5	91.62	94.10	95.05
0.6	91.14	93.71	95.33
0.7	90.10	93.81	94.86
0.8	90.00	79.90	95.14
0.9	89.71	68.00	95.24

Random Forest

Random Forest (RF) was proposed by Breiman (Lou et al., 2014; Sitokonstantinou et al., 2018), is an ensemble learning method. The RF algorithm generates a large number of decision trees, in which every single tree produces classification results and then merged all the results of all decision trees using the voting method to generate the final result (Jiang et al., 2007a; Sitokonstantinou et al., 2018). Feature selection in RF is random, i.e., it is not using all the features; it divides the features into different trees and then merges the final result of every tree (Jiang et al., 2007a). Two parameters that are ntree and mtry are needed to be set up for achieving better accuracy. The ntree is the number of trees, while the mtry is the number of samples/variables in each split (Noi and Kappas, 2018). The RF algorithm produces better performance on a large dataset; however, it experience with overfitting in case the dataset is too noisy.

PERFORMANCE METRICS

The performance of a newly constructed predictor based on statistical machine learning algorithms can be evaluated through some procedures before it applies in a real production environment (Baratloo et al., 2015). However, before moving forward, we need to consider the following two questions: (a) what measurement metrics should be adopted to evaluate the performance of a new predictor? (b) what test approach should be employed to compute the measurement metrics? Several metrics have been proposed in the literature for performance evaluation of a machine learning model (Chou, 2001a,b; Xu et al., 2013; Lin et al., 2014; Zhang et al., 2016; Liu et al., 2017c, 2018; Feng et al., 2019; Tahir et al., 2019). In all these metrics, accuracy is considered as the most eminent metric for the performance measurement of a model, however, only the accuracy cannot be sufficient to assess the model significance (Guo et al., 2014; Akbar and Hayat, 2018). Therefore, a set of four different measurement metrics along with the accuracy were considered to evaluate the performance of a predictor. These metrics are: (i) overall accuracy (ACC), (ii) sensitivity (SN), (iii) specificity (SP), and (iv)

Mathew's correlation coefficient (MCC). The ACC determines a ratio of number of corrected predictions made by the model to the total number of input samples. The SN returns true positive rate of a model whereas the SP is opposite of the SN and compute true negative rate of a model. The MCC uses all positive and negative instances and produces output in the range of +1 and −1. Further, the details and meanings of these metrics are clearly mentioned in series of publications (e.g., see Chen et al., 2007, 2013; Guo et al., 2008; Qiu et al., 2014; Kabir and Hayat, 2016; Sabooh et al., 2018; Zhang and Kong, 2018b; Khan et al., 2019b; Raza, 2019).

In this paper, we considered the aforementioned four metrics to assess the outcomes of the proposed iRSpot-DNN for the prediction of recombination spots. According to Chou's symbol studying in signal protein peptides (Chou, 2001a), the four metrics can be represented in the following equations in order to make them easily understandable to most experimental scientists.

$$\mathrm{ACC} = 1 - \frac{H_{-}^{+} + H_{+}^{-}}{H^{+} + H^{-}};\ 0 \leq \mathrm{ACC} \leq 1 \tag{12}$$

$$\mathrm{SN} = 1 - \frac{H_{-}^{+}}{H^{+}};0 \leq \mathrm{SN} \leq 1 \tag{13}$$

$$\mathrm{SP} = 1 - \frac{H_{+}^{-}}{H^{-}};\ 0 \leq \mathrm{SP} \leq 1 \tag{14}$$

$$\mathrm{MCC} = \frac{1 - \left(\frac{H_{-}^{+} + H_{+}^{-}}{H^{+} + H^{-}}\right)}{\sqrt{\left(1 + \frac{H_{+}^{-} - H_{-}^{+}}{H^{+}}\right)\left(1 + \frac{H_{-}^{+} - H_{+}^{-}}{H^{-}}\right)}};\ -1 \leq \mathrm{MCC} \leq 1 \tag{15}$$

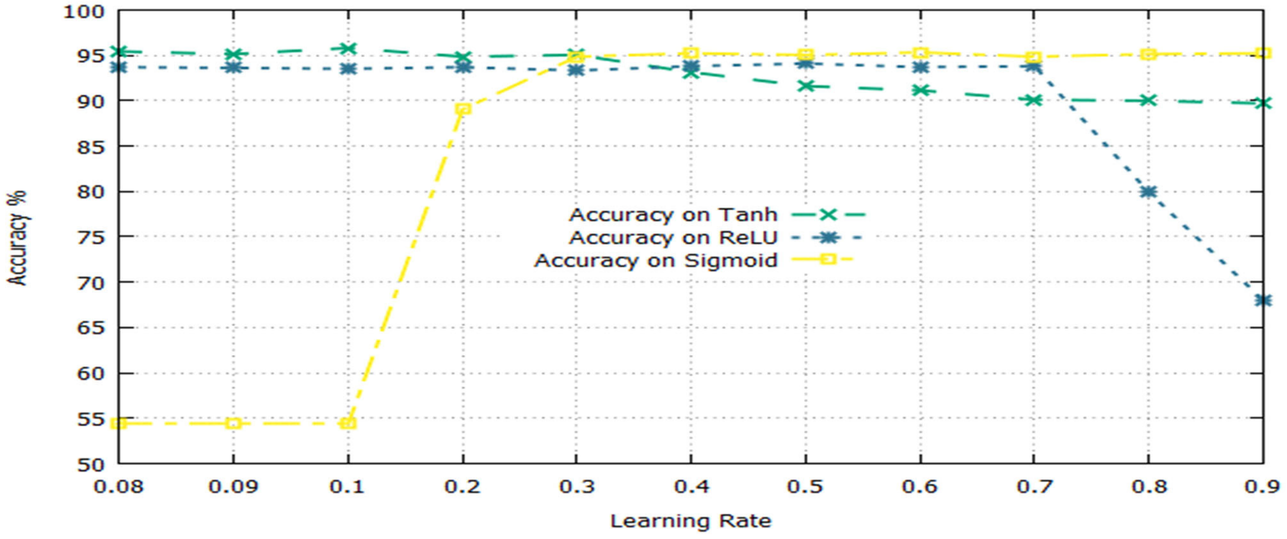

FIGURE 3 | Impact of learning rates and activation function on the performance of DNN.

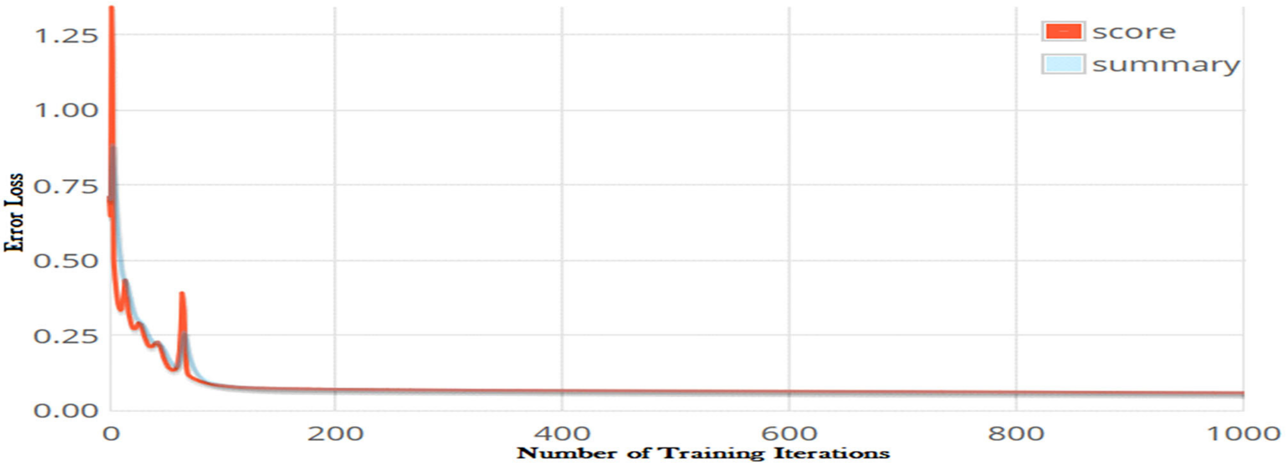

FIGURE 4 | Error losses of DNN model on different iterations using Tanh activation function.

In the above equations, H^{+} represents the total number of hotspots, H^{-} represents the number of coldspots. Similarly H_{-}^{+} represents the number of hotspots wrongly predicted as coldspots and H_{+}^{-} represents the number of coldspots that are incorrectly predicted as hotspots.

The next challenge is how to assess the quality of a new predictor using the metrics values. For this purpose, three methods, such as jackknife, independent dataset, and K-fold cross-validation are widely applied in the literature (Chen et al., 2013; Liu et al., 2017a; Yang et al., 2018; Zhang and Kong, 2018a,b; Kong and Zhang, 2019) to examine the performance and robustness of a predictor. It is to be noted that in the cited literature, there is no independent dataset is available so for, that is why we are unable to apply independent dataset method whereas the jackknife method is computationally expensive because of its working mechanism. Hence, In this study, we employed K-fold cross-validation (i.e., K = 10) method as it has been adopted by several investigators to assess the quality of their predictors (Zhou et al., 2006; He et al., 2018; Kong and Zhang, 2019) and comparatively less time consuming technique compare with jackknife method.

TABLE 7 | Performance of DNN Model using different feature extraction methods.

Groups/Methods	SN (%)	SP (%)	ACC (%)	MCC
Gap	86.68	79.60	82.29	0.6458
Reverse	82.87	80.58	81.52	0.6270
PseTNC	75.78	77.17	76.57	0.5261
G1	94.39	95.78	95.14	0.9021
G2	83.73	85.08	84.48	0.6866
G3	96.17	95.52	95.81	0.9155
G4	86.65	86.17	86.38	0.7250
G5	94.15	95.27	94.76	0.8944
G6	91.18	92.33	91.81	0.8348
G7	93.25	94.08	93.70	0.8729
G8	87.58	86.14	86.76	0.7328
G9	90.85	92.79	91.90	0.8369

EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we discuss the efficiency and significance of the proposed iRSpot-DNN. Firstly, we discuss the hyper-parameter optimization using a grid search technique. Secondly, we assess the performance of the DNN algorithm using various sequence formulation techniques along with the hybrid features. Thirdly, the performance of the DNN is compared with other machine learning algorithms. Finally, the outcome of the proposed iRSpot-DNN is compared with recently published models.

Hyper-Parameter Optimization

Deep learning algorithms consider a number of hyper-parameters during model configuration. The configurations of these parameters have a significant impact on the performance of a learning algorithm. Unlink regular parameters, the hyper-parameters are specified by user during the model setup. Typically, it is a challenging task to know what value to be configured for the hyper-parameters of a learning model on

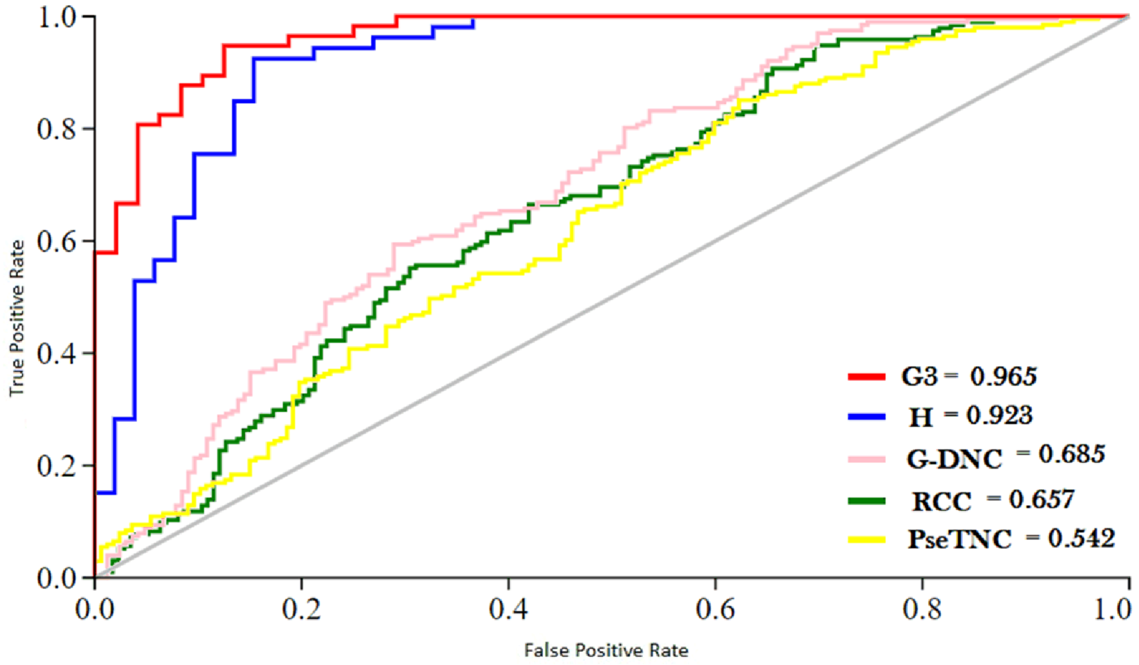

FIGURE 5 | ROC of DNN model using different feature group.

a given dataset. Therefore, different approaches, i.e., manual trail, grid search (Fowler, 2000), and random search are commonly used for hyper-parameter tuning. The manual trail and random search approaches for hyper-parameter tuning are laborious, time-consuming and un-methodical. Hence, we apply a grid search technique to find optimal hyper-parameters for the proposed model. In order to apply the grid search approach, we build a model for different combinations of hyper-parameters, evaluate the model for every combination and store the results. The set of hyper-parameters that gives the best result amongst all combinations is selected and considered as the optimal parameter set for the proposed model. During the hyper-parameter optimization, we consider the most influential parameters, such as activation function, learning rate, and a number of iterations. We examined the model hyper-parameter on all groups of features using grid search approach, however, the promising results were obtained on G3 group features. The optimal hyper-parameters obtained through the grid search method for the proposed model are presented in **Table 5**.

The learning rate in machine learning algorithms is a vital component and determines the step size a model takes at each iteration. The step size is the amount that weights updated during the model training phase. The value of the learning rate can be a small positive value in the range between 0.0 and 1.0. A small value of the learning rate may lead to overfitting and takes a longer time to train the model, whereas a large value can quickly train the model. However, it may ignore some best characteristics of features being used during the model training.

TABLE 8 | Performance comparison of DNN with other classifiers.

Classification algorithm	Feature method	ACC (%)	SN (%)	SP (%)	MCC
SVM	Gapped di-nucleotide composition	82.67	72.80	90.91	0.6534
	Reverse complement composition	54.48	–	–	–
	PseTri-nucleotide composition	80.19	70.08	88.64	0.6022
	H	93.05	92.68	93.36	0.860
	G3	92.95	92.68	93.18	0.858
KNN	Gapped di-nucleotide composition	74.67	47.91	94.03	0.5283
	Reverse complement composition	69.90	35.98	95.25	0.4504
	PseTri-nucleotide composition	74.86	48.12	95.20	0.5329
	H	75.24	71.13	78.67	0.499
	G3	75.43	71.34	78.85	0.5035
RF	Gapped di-nucleotide composition	80.19	74.69	84.79	0.5996
	Reverse complement composition	80.29	75.94	83.92	0.6015
	PseTri-nucleotide composition	81.71	75.73	86.71	0.6307
	H	84.57	80.75	87.76	0.688
	G3	84.57	78.87	89.34	0.6888
DNN	Gapped di-nucleotide composition	82.29	86.68	79.60	0.6458
	Reverse complement composition	81.52	82.87	80.58	0.6270
	PseTri-nucleotide composition	76.57	75.78	77.17	0.5261
	H	95.14	94.39	95.52	0.9021
	G3	95.81	96.17	95.78	0.9155

The activation function is a significant component of a deep learning algorithm. It is a non-linear function employed at a neuron and computes the output of a hidden layer. The activation function decides either a neuron should be fired or ignored based on the information computed at a hidden layer. Different activation functions can be applied in deep learning algorithms, however, the commonly applied activation functions are: sigmoid, Tanh, and Rectified Linear Unit (ReLU). The impact of both the learning rate and activation functions are reported in **Table 6** and illustrated in **Figure 3**.

Table 6 shows that the DNN model achieved a highest accuracy, i.e., 95.81% using Tanh with a combination of learning rate 0.1. The second highest accuracy, i.e., 95.33% reported the sigmoid with a combination of learning rate 0.6. The ReLU yielded the third-highest success rate, i.e., 94.10% with a combination of learning rate 0.5. Furthermore, it can be noted from **Figure 3**, the un-tune parameter can significantly affect the performance and stability of the model. Moreover, the model shows a stable performance using Tanh compared with other activation functions. Hence, the Tanh can be considered an optimum value for the activation function parameter.

The number of iteration is another optimization parameter and significantly impacts the performance of a learning model. Increasing the number of iterations can significantly minimize the loss function of a model; however, it may increases the model training time considerably. The error loss of the DNN model on different iterations is shown in **Figure 4**. It can be observed from the figure that increasing the number of iterations significantly reduced loss function. The minimum error loss, i.e., 0.0002176 was reported at iteration 1,000 as shown in the figure. We further increased the number of iterations; however, it did not significantly affect the loss function.

Performance of DNN Model Using Different Feature Extraction Methods

The performance of the proposed DNN model was analyzed using different feature extraction methods along with the hybrid features, as discussed in section Sequence Formulation Methods. The analysis results are presented in **Table 7**. From **Table 7**, we can observe that the DNN model accomplished a highest accuracy of 95.81% using features represented in the G3 group whereas, the lowest accuracy achieved using PseTNC method.

ROC (Receiver Operating characteristic)/AUC (Area Under ROC) curve is another effective method that measure the quality of a prediction model. We evaluated the performance of the DNN model on different feature extraction groups using ROC curve as shown in **Figure 5**. The figure shows that the DNN model generated a highest value, i.e., 0.965 of ROC/AUC using G3 group features compared with others groups features.

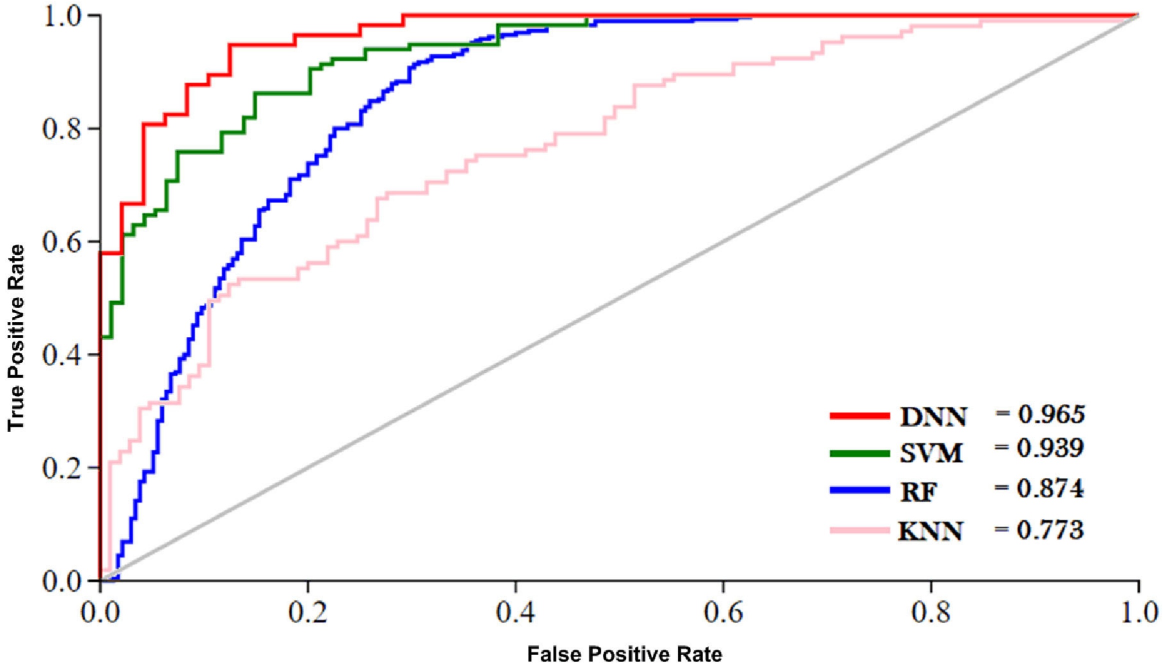

FIGURE 6 | ROC of different classifiers.

TABLE 9 | Comparison of the proposed model with existing predictors.

Methods	SN (%)	SP (%)	MCC	ACC (%)
RF-DYMHC (Jiang et al., 2007b)	73.01	86.56	0.6049	80.40
IDQD (Liu et al., 2012)	79.52	81.82	0.6160	80.77
iRSpot-PseDNC (Chen et al., 2013)	71.75	85.84	0.5830	79.33
iRSpot-TNCPseAAC (Qiu et al., 2014)	76.56	70.99	0.4737	73.52
iRSpot-DACC (Liu et al., 2016)	75.71	88.16	0.6470	82.52
iRSpot-EL (Liu et al., 2017a)	75.29	88.81	0.6510	82.65
iRSpot-ADPM (Zhang and Kong, 2018a)	77.19	90.73	0.6905	84.57
iRSpot-SPI (Khan et al., 2019b)	92.21	92.11	0.8101	90.04
iRecSpot-EF (Jani et al., 2018)	95.14	95.80	0.9037	95.14
Proposed iRSpot-DNN	96.17	95.89	0.9155	95.81

Comparison of DNN Model With Different Machine Learning Algorithms

The outcome of the DNN in comparison with different learning classifiers, including SVM (Yue et al., 2003), KNN (Cheng et al., 2014), and RF (Fawagreh et al., 2014) is presented in this section. We employed 10-fold cross-validation tests to assess the outcome of the classifiers using different sequence formulation methods. The results of this comparison are shown in **Table 8**. The table shows that the DNN model achieved a highest accuracy, i.e., 95.81% compared with other machine learning algorithms. The second highest accuracy (i.e., 93.05) reported by the SVM and third-ranking accuracy (i.e., 84.57) obtained by the RF algorithm. The KNN classifier achieved the lowest accuracy (i.e., 75.43). Additionally, we evaluated the performance of the classifiers in more comprehensive way using ROC curve. For the ROC curve, we used only G3 group features for all the classifiers because they generated promising results using G3 group features as shown in **Table 8**. The ROC curve of the classifiers is shown in **Figure 6**. From the figure we can observe that the proposed DNN model generated a highest ROC/AUC value, i.e., 0.965 compared with the ROC/AUC values of the other classifiers.

Comparison of the Proposed Predictor With Existing Predictors

This section presents a performance comparison of the proposed predictor with the existing predictors. For the comparison, we selected 9 recently published predictors from the literature. These predictors are: RF-DYMHC (Jiang et al., 2007b), IDQD (Liu et al., 2012), iRSpot-PseDNC (Chen et al., 2013), iRSpotTNCPseAAC (Qiu et al., 2014), iRSpot-DACC (Liu et al., 2016), iRSpot-EL (Liu et al., 2017a), iRSpot-ADPM (Zhang and Kong, 2018a), iRSpot-SPI (Khan et al., 2019b), and iRecSpot-EF (Jani et al., 2018). For this comparison, the proposed model utilized G3 group features. Results of this comparison in terms of accuracy, sensitivity, specificity, and MMC are listed in **Table 9**. It is evidently presented in **Table 9** that the proposed model outperformed the existing models in terms of all four metrics.

For example, consider the MCC metric, the proposed model achieved the highest value, i.e., 0.9155, the second highest value, i.e., 0.9037 reported by the iRecSpot-EF and the third highest value, i.e., 0.8101 generated by the iRSpot-SPI. Similarly, in case of the accuracy metric, the iRecSpot-EF generated 95.14% accuracy whereas, the proposed iRSpot-DNN reported 95.81% accuracy and the iRSpot-SPI produced 90.04% accuracy. These results confirmed that the proposed model outperformed the existing models and can predict the recombination spots with high precision.

CONCLUSION AND FUTURE WORK

This study presented an intelligent computation model for the identification of recombination spots. In the proposed model, hybrid features were extracted from the benchmark dataset. A novel formula was derived for feature selection and it has 2-fold advantages. Firstly, it avoids biasness among different selected feature extraction algorithms. Secondly, it preserved the sequence discriminant properties along with the sequence-structure information. The performance of the proposed model was investigated on different classifiers. The results exhibit that the optimized deep neural network obtained higher performance having accuracy of 95.81%. As compared to the existing methods the proposed model performance on the prediction of recombination spots is clearly improved. It is realized that the proposed predictor can be considered a handy identification tool and potentially apply to basic research and drug discovery. In future work, we will design a public web server for the proposed iRSpot-DNN so that every experimental scientist can easily access and use for identification of recombination spots.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

FK, MK, and NI contributed in the main idea of this research, the model concept design, and supervised. FK and SK contributed the code development and conducted experiments. FK wrote the first draft of the manuscript. MK, NI, and DM performed the statistical analysis. MK, NI, AK, and D-QW contributed the results analysis and manuscript revision. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

D-QW was supported by the grants from the Key Research Area Grant No. 2016YFA0501703 of the Ministry of Science and Technology of China, the National Natural Science Foundation of China (Contract Nos. 61832019, 61503244), the Science and Technology Commission of Shanghai Municipality (Grant No. 19430750600), the Natural Science Foundation of Henan Province (162300410060), and Joint Research Funds for Medical and Engineering and Scientific Research at Shanghai Jiao Tong University (YG2017ZD14). The computations were partially performed at the Pengcheng Lab and the Center for High-Performance Computing, Shanghai Jiao Tong University.

REFERENCES

Abeysinghe, S. S., Chuzhanova, N., Krawczak, M., Ball, E. V., and Cooper, D. N. (2003). Translocation and gross deletion breakpoints in human inherited disease and cancer I: Nucleotide composition and recombination-associated motifs. *Hum. Mutat.* 22, 229–244. doi: 10.1002/humu.10254

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., and Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.* 100, 270–278. doi: 10.1016/j.compbiomed.2017.09.017

Ahmad, S., Kabir, M., and Hayat, M. (2015). Identification of heat shock protein families and J-protein types by incorporating dipeptide composition into Chou's general PseAAC. *Comput. Methods Programs Biomed.* 122, 165–174. doi: 10.1016/j.cmpb.2015.07.005

Akbar, S., and Hayat, M. (2018). iMethyl-STTNC: Identification of N6-methyladenosine sites by extending the idea of SAAC into Chou's PseAAC to formulate RNA sequences. *J. Theor. Biol.* 455, 205–211. doi: 10.1016/j.jtbi.2018.07.018

Ali, Z., Abbas, A. W., Thasleema, T. M., Uddin, B., Raaz, T., and Abid, S. A. R. (2015). Database development and automatic speech recognition of isolated Pashto spoken digits using MFCC and K-NN. *Int. J. Speech Technol.* 18, 271–275. doi: 10.1007/s10772-014-9267-z

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Mol. Syst. Biol.* 12:878. doi: 10.15252/msb.20156651

Ballanti, L., Blesius, L., Hines, E., and Kruse, B. (2016). Tree species classification using hyperspectral imagery: a comparison of two classifiers. *Rem. Sens.* 8, 1–18. doi: 10.3390/rs8060445

Baratloo, A., Hosseini, M., Negida, A., and El Ashal, G. (2015). Part 1: simple definition and calculation of accuracy, sensitivity and specificity. *Emerg. (Tehran)* 3, 48–9.

Bordes, A., and Weston, J. (2014). "Question answering with subgraph embeddings," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha), 615–620. doi: 10.3115/v1/D14-1067

Cai, Y. D., Zhou, G. P., and Chou, K. C. (2003). Support vector machines for predicting membrane protein types by using functional domain composition. *Biophys. J.* 8, 3257–3263. doi: 10.1016/S0006-3495(03)70050-2

Chen, J., Liu, H., Yang, J., and Chou, K. C. (2007). Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. *Amino Acids* 33, 423–428. doi: 10.1007/s00726-006-0485-9

Chen, W., Feng, P. M., Lin, H., and Chou, K. C. (2013). IRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic Acids Res.* 41, 1–9. doi: 10.1093/nar/gks1450

Chen, Z., Li, F., Zhao, P., Marquez-lago, T., and Leier, A. (2019). iLearn, an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Br. Bioinform.* 21, 1047–1057. doi: 10.1093/bib/bbz041

Cheng, D., Zhang, S., Deng, Z., Zhu, Y., and Zong, M. (2014). "k NN algorithm with data-driven k value," in *International Conference on Advanced Data Mining and Applications* (Guilin), 499–512. doi: 10.1007/978-3-319-14717-8_39

Chou, K.-C. (2015). Impacts of bioinformatics to medicinal chemistry. *Med. Chem.* 11, 218–234. doi: 10.2174/1573406411666141229162834

Chou, K. C. (2001a). Using subsite coupling to predict signal peptides. *Protein Eng. Des. Sel.* 14, 75–79. doi: 10.1093/protein/14.2.75

Chou, K. C. (2001b). Prediction of protein signal sequences and their cleavage sites. *Proteins Struct. Funct. Genet.* 42, 136–139. doi: 10.1002/1097-0134(20010101)42:1<136::AID-PROT130>3.0.CO;2-F

Chou, K. C., and Elrod, D. W. (2002). Bioinformatical analysis of G-protein-coupled receptors. *J. Proteome Res.* 1, 429–433. doi: 10.1021/pr025527k

Chou, K. C., Wu, Z. C., and Xiao, X. (2011). iLoc-Euk: a multi-label classifier for predicting the subcellular localization of singleplex and multiplex eukaryotic proteins. *PLoS ONE* 6:e18258. doi: 10.1371/journal.pone.0018258

Cohn, D., Zuk, O., and Kaplan, T. (2018). Enhancer identification using transfer and adversarial deep learning of DNA sequences. *bioRXiv*. doi: 10.1101/264200

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297. doi: 10.1007/BF00994018

Couprie, C., Najman, L., and Lecun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1915–1929. doi: 10.1109/TPAMI.2012.231

Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. In *Modern Speech Recognition*, Vol. 29 (IEEE) 82–97.

Donaldson, R. W. (1967). Approximate formulas for the information transmitted by a discrete communication channel. *IEEE Trans. Inf. Theory* 13, 118–119. doi: 10.1109/TIT.1967.1053945

Dong, C., Yuan, Y., Zhang, F., and Hua, H. (2016). Molecular BioSystems Combining pseudo dinucleotide composition with the Z curve method to improve the accuracy of predicting DNA elements : a case study in recombination spots. *Mol. Biosyst.* 12, 2893–2900. doi: 10.1039/C6MB00374E

Du, P., Wang, X., Xu, C., and Gao, Y. (2012). PseAAC-Builder: a cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions. *Anal. Biochem.* 425, 117–119. doi: 10.1016/j.ab.2012.03.015

Dwivedi, A. K. (2018). Artificial neural network model for effective cancer classification using microarray gene expression data. *Neural Comput. Appl.* 29, 1545–1554. doi: 10.1007/s00521-016-2701-1

Fawagreh, K., Gaber, M. M., and Elyan, E. (2014). Random forests: from early developments to recent advancements. *Syst. Sci. Control Eng.* 2, 602–609. doi: 10.1080/21642583.2014.956265

Feng, P., Yang, H., Ding, H., Lin, H., Chen, W., and Chou, K. C. (2019). iDNA6mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. *Genomics* 111, 96–102. doi: 10.1016/j.ygeno.2018.01.005

Fowler, B. (2000). A sociological analysis of the satanic verses affair. *Theory Cult. Soc.* 17, 39–61. doi: 10.1177/02632760022050997

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). CD-HIT: Accelerated for clustering the next-generation sequencing data. *Bioinformatics* 28, 3150–3152. doi: 10.1093/bioinformatics/bts565

Ghandi, M., Lee, D., Mohammad-Noori, M., and Beer, M. A. (2014). Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput. Biol.* 10:e1003711. doi: 10.1371/journal.pcbi.1003711

Guo, S. H., Deng, E. Z., Xu, L. Q., Ding, H., Lin, H., Chen, W., et al. (2014). INuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics* 30, 1522–1529. doi: 10.1093/bioinformatics/btu083

Guo, Y., Yu, L., Wen, Z., and Li, M. (2008). Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Res.* 36, 3025–3030. doi: 10.1093/nar/gkn159

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46, 389–422. doi: 10.1023/A:1012487302797

Harrison, O. (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Towards Data Science. Available online at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 (accessed February 16, 2020).

He, W., Ju, Y., Zeng, X., Liu, X., and Zou, Q. (2018). Sc-ncDNA pred: a sequence-based predictor for identifying non-coding DNA in *Saccharomyces cerevisiae*. *Front. Microbiol.* 9:2174. doi: 10.3389/fmicb.2018.02174

He, X., Han, K., Hu, J., Yan, H., Yang, J. Y., Shen, H. Bin, et al. (2015). TargetFreeze: identifying antifreeze proteins via a combination of weights using sequence evolutionary information and pseudo amino acid composition. *J. Membr. Biol.* 248, 1005–1014. doi: 10.1007/s00232-015-9811-z

Hey, J. (2004). What's so hot about recombination hotspots? *PLoS Biol.* 2:e190. doi: 10.1371/journal.pbio.0020190

Hu, L. Y., Huang, M. W., Ke, S. W., and Tsai, C. F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *Springerplus* 5:1304. doi: 10.1186/s40064-016-2941-7

Jani, R., Mozlish, T. K., Ahmed, S., Farid, D., and Shatabda, S. (2018). iRecSpot-EF: effective sequence based features for recombination hotspot prediction. *Comput. Biol. Med.* 103, 17–23. doi: 10.1016/j.compbiomed.2018.10.005

Jia, J., Liu, Z., Xiao, X., Liu, B., and Chou, K.-C. (2015). iPPI-Esml: An ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC. *J. Theor. Biol.* 377, 47–56. doi: 10.1016/j.jtbi.2015.04.011

Jiang, P., Wu, H., Wang, W., Ma, W., Sun, X., and Lu, Z. (2007a). MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res.* 35, 339–344. doi: 10.1093/nar/gkm368

Jiang, P., Wu, H., Wei, J., Sang, F., Sun, X., and Lu, Z. (2007b). RF-DYMHC: detecting the yeast meiotic recombination hotspots and coldspots by random forest model using gapped dinucleotide composition features. *Nucleic Acids Res.* 35, 47–51. doi: 10.1093/nar/gkm217

Kabir, M., and Hayat, M. (2016). iRSpot-GAEnsC: identifing recombination spots via ensemble classifier and extending the concept of Chou's PseAAC to formulate DNA samples. *Mol. Genet. Genomics* 291, 285–296. doi: 10.1007/s00438-015-1108-5

Kabir, M., and Yu, D. J. (2017). Predicting DNase I hypersensitive sites via un-biased pseudo trinucleotide composition. *Chemom. Intell. Lab. Syst.* 167, 78–84. doi: 10.1016/j.chemolab.2017.05.001

Kelley, D. R., Snoek, J., and Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* 26, 990–999. doi: 10.1101/gr.200535.115

Khan, M., Hayat, M., Khan, S. A., Ahmad, S., and Iqbal, N. (2017). Bi-PSSM: position specific scoring matrix based intelligent computational model for identification of mycobacterial membrane proteins. *J. Theor. Biol.* 435, 116–124. doi: 10.1016/j.jtbi.2017.09.013

Khan, S., Khan, M., Iqbal, N., Hussain, T., Khan, S. A., and Chou, K.-C. (2019a). A two-level computation model based on deep learning algorithm for identification of piRNA and their functions via Chou's 5-steps rule. *Int. J. Pept. Res. Ther.* 26, 795–809. doi: 10.1007/s10989-019-09887-3

Khan, S., Khan, M., Iqbal, N., Khan, S. A., and Chou, K.-C. (2020). Prediction of piRNAs and their function based on discriminative intelligent model using hybrid features into Chou's PseKNC. *Chemom. Intell. Lab. Syst.* 203:104056. doi: 10.1016/j.chemolab.2020.104056

Khan, Z. U., Ali, F., Khan, I. A., Hussain, Y., and Pi, D. (2019b). iRSpot-SPI: deep learning-based recombination spots prediction by incorporating secondary sequence information coupled with physio-chemical properties via Chou's 5-step rule and pseudo components. *Chemom. Intell. Lab. Syst.* 189, 169–180. doi: 10.1016/j.chemolab.2019.05.003

Kondarasaiah, M. H., and Ananda, S. (2004). Kinetic and mechanistic study of Ru(III)-nicotinic acid complex formation by oxidation of bromamine-T in acid solution. *Oxidat. Commun.* 27, 140–147.

Kong, L., and Zhang, L. (2019). I6mA-DNCP: computational identification of DNA N6-methyladenine sites in the rice genome using optimized dinucleotide-based features. *Genes (Basel).* 10:828. doi: 10.3390/genes10100828

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems–Volume 1* NIPS'12 (Curran Associates Inc.), 1097–1105. Available online at: http://dl.acm.org/citation.cfm?id=2999134.2999257

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Leung, M. K. K., Xiong, H. Y., Lee, L. J., and Frey, B. J. (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30, 121–129. doi: 10.1093/bioinformatics/btu277

Li, G. Q., Liu, Z., Shen, H. Bin, and Yu, D. J. (2016). TargetM6A: identifying N6-methyladenosine sites from RNA sequences via position-specific nucleotide propensities and a support vector machine. *IEEE Trans. Nanobiosci*. 15, 674–682. doi: 10.1109/TNB.2016.2599115

Li, L., Yu, S., Xiao, W., Li, Y., Huang, L., Zheng, X., et al. (2014). Sequence-based identification of recombination spots using pseudo nucleic acid representation and recursive feature extraction by linear kernel SVM. *BMC Bioinformatics* 15:340. doi: 10.1186/1471-2105-15-340

Li, W., and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 1658–1659. doi: 10.1093/bioinformatics/btl158

Lichten, M., and Goldman, A. S. H. (1995). Meiotic recombination hotspots. *Annu. Rev. Genet.* 29, 423–444. doi: 10.1146/annurev.ge.29.120195.002231

Lin, H., Deng, E. Z., Ding, H., Chen, W., and Chou, K. C. (2014). IPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res.* 42, 12961–12972. doi: 10.1093/nar/gku1019

Liu, B., Liu, F., Wang, X., Chen, J., Fang, L., and Chou, K. C. (2015). Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res.* 43, W65–W71. doi: 10.1093/nar/gkv458

Liu, B., Liu, Y., Jin, X., Wang, X., and Liu, B. (2016). IRSpot-DACC: a computational predictor for recombination hot/cold spots identification based on dinucleotide-based auto-cross covariance. *Sci. Rep.* 6, 1–9. doi: 10.1038/srep33483

Liu, B., Wang, S., Long, R., and Chou, K. C. (2017a). IRSpot-EL: Identify recombination spots with an ensemble learning approach. *Bioinformatics* 33, 35–41. doi: 10.1093/bioinformatics/btw539

Liu, B., Wu, H., and Chou, K.-C. (2017b). Pse-in-One 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nat. Sci.* 9, 67–91. doi: 10.4236/ns.2017.94007

Liu, B., Yang, F., and Chou, K. C. (2017c). 2L-piRNA: a two-layer ensemble classifier for identifying Piwi-interacting RNAs and their function. *Mol. Ther. Nucleic Acids* 7, 267–277. doi: 10.1016/j.omtn.2017.04.008

Liu, B., Yang, F., Huang, D. S., and Chou, K. C. (2018). IPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics*. 34, 33–40. doi: 10.1093/bioinformatics/btx579

Liu, G., Liu, J., Cui, X., and Cai, L. (2012). Sequence-dependent prediction of recombination hotspots in Saccharomyces cerevisiae. *J. Theor. Biol.* 293, 49–54. doi: 10.1016/j.jtbi.2011.10.004

Lopez, P., Philippe, H., Myllykallio, H., and Forterre, P. (1999). Identification of putative chromosomal origins of replication in archaea. *Mol. Microbiol.* 32, 883–886. doi: 10.1046/j.1365-2958.1999.01370.x

Lou, W., Wang, X., Chen, F., Chen, Y., Jiang, B., and Zhang, H. (2014). Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naïve Bayes. *PLoS ONE* 9:e86703. doi: 10.1371/journal.pone.0086703

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). Deep neural nets as a method for quantitative structure–activity relationships. *J. Chem. Inf. Model.* 55, 263–274. doi: 10.1021/ci500747n

Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Mol. Pharm.* 13, 1445–1454. doi: 10.1021/acs.molpharmaceut.5b00982

Maruf, A. Al., and Shatabda, S. (2018). Genomics iRSpot-SF: prediction of recombination hotspots by incorporating sequence based features into Chou's pseudo components. *Genomics* 111, 966–974. doi: 10.1016/j.ygeno.2018.06.003

Miao, J. H., and Miao, K. H. (2018). Cardiotocographic diagnosis of fetal health based on multiclass morphologic pattern predictions using deep learning classification. *Int. J. Adv. Comput. Sci. Appl.* 9, 1–11. doi: 10.14569/IJACSA.2018.090501

Mikolov, T., Kombrink, S., Burget, L., Cernocký, J., and Khudanpur, S. (2011). "Extensions of recurrent neural network language model," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Prague), 5528–5531. doi: 10.1109/ICASSP.2011.5947611

Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Brief. Bioinform.* 18, 851–869. doi: 10.1093/bib/bbw068

Noi, P. T., and Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors* 18:18. doi: 10.3390/s18010018

Paul, P., Nag, D., and Chakraborty, S. (2016). Recombination hotspots: models and tools for detection. *DNA Repair (Amst).* 40, 47–56. doi: 10.1016/j.dnarep.2016.02.005

Petes, T. D. (2001). Meiotic recombination hot spots and cold spots. *Nat. Rev. Genet.* 2, 360–369. doi: 10.1038/35072078

Qian, Y., Zhou, W., Yan, J., Li, W., and Han, L. (2015). Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. *Rem. Sens.* 7, 153–168. doi: 10.3390/rs70100153

Qin, Z., Wang, A. T., Zhang, C., and Zhang, S. (2013). "Cost-sensitive classification with k-nearest neighbors," in *Knowledge Science, Engineering and Management. KSEM 2013. Lecture Notes in Computer Science*, Vol. 8041, ed M. Wang (Berlin; Heidelberg: Springer), 112–131. doi: 10.1007/978-3-642-39787-5_10

Qiu, W. R., Xiao, X., and Chou, K. C. (2014). iRSpot-TNCPseAAC: identify recombination spots with trinucleotide composition and pseudo amino acid components. *Int. J. Mol. Sci.* 15, 1746–1766. doi: 10.3390/ijms15021746

Quang, D., and Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* 44, 1–6. doi: 10.1093/nar/gkw226

Raza, K. (2019). *Improving the Prediction Accuracy of Heart Disease With Ensemble Learning and Majority Voting Rule*. Elsevier Inc.

Sabooh, M. F., Iqbal, N., Khan, M., Khan, M., and Maqbool, H. F. (2018). Identifying 5-methylcytosine sites in RNA sequence using composite encoding feature into Chou's PseKNC. *J. Theor. Biol.* 452, 1–9. doi: 10.1016/j.jtbi.2018.04.037

Sainath, T. N., Mohamed, A., Kingsbury, B., Ramabhadran, B., Watson, I. B. M. T. J., and Heights, Y. (2013). "Deep convolutional neural network for LVCSR," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 10–14. Available at: http://www.cs.toronto.edu/~asamir/papers/icassp13_cnn.pdf (accessed February 16, 2020).

Sitokonstantinou, V., Papoutsis, I., Kontoes, C., Arnal, A. L., Andrés, A. P. A., and Zurbano, J. A. G. (2018). Scalable parcel-based crop identification scheme using Sentinel-2 data time-series for the monitoring of the common agricultural policy. *Rem. Sens.* 10, 5–32. doi: 10.3390/rs10060911

Tahir, M., Tayara, H., and Chong, K. T. (2019). iPseU-CNN: identifying RNA pseudouridine sites using convolutional neural networks. *Mol. Ther. Nucleic Acids* 16, 463–470. doi: 10.1016/j.omtn.2019.03.010

Tang, H., Zou, P., Zhang, C., Chen, R., Chen, W., and Lin, H. (2016). Identification of apolipoprotein using feature selection technique. *Sci. Rep.* 6, 1–6. doi: 10.1038/srep30441

Telenti, A., Lippert, C., Chang, P. C., and DePristo, M. (2018). Deep learning of genomic variation and regulatory network data. *Hum. Mol. Genet.* 27, R63–R71. doi: 10.1093/hmg/ddy115

Tompson, J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Adv. Neural Inf. Process. Syst.* 2, 1799–1807.

Van Der Malsburg, C. (1986). "Frank Rosenblatt: principles of neurodynamics: perceptrons and the theory of brain mechanisms," in *Brain Theory*, eds G. Palm and A. Aertsen (Berlin; Heidelberg: Springer), 245–248. doi: 10.1007/978-3-642-70911-1_20

Wang, R., Xu, Y., and Liu, B. (2016). Recombination spot identification Based on gapped k-mers. *Sci. Rep.* 6:23934. doi: 10.1038/srep35331

Xu, Y., Shao, X. J., Wu, L. Y., Deng, N. Y., and Chou, K. C. (2013). ISNO-AAPair: Incorporating amino acid pairwise coupling into PseAAC for predicting cysteine S-nitrosylation sites in proteins. *PeerJ* 2013, 1–18. doi: 10.7717/peerj.171

Yang, H., Qiu, W. R., Liu, G., Guo, F. B., Chen, W., Chou, K. C., et al. (2018). IRSpot-Pse6NC: Identifying recombination spots in *Saccharomyces cerevisiae* by incorporating hexamer composition into general PseKNC. *Int. J. Biol. Sci.* 14, 883–891. doi: 10.7150/ijbs.24616

Yue, S., Li, P., and Hao, P. (2003). SVM classification:Its contents and challenges. *Appl. Math. J. Chinese Univ.* 18, 332–342. doi: 10.1007/s11766-003-0059-5

Zavaljevski, N., Stevens, F. J., and Reifman, J. (2002). Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics* 18, 689–696. doi: 10.1093/bioinformatics/18.5.689

Zhang, C. J., Tang, H., Li, W. C., Lin, H., Chen, W., and Chou, K. C. (2016). iOri-Human: identify human origin of replication by incorporating dinucleotide physicochemical properties into pseudo nucleotide composition. *Oncotarget* 7, 69783–69793. doi: 10.18632/oncotarget.11975

Zhang, L., and Kong, L. (2018a). iRSpot-ADPM: identify recombination spots by incorporating the associated dinucleotide product model into Chou's pseudo components. *J. Theor. Biol.* 441, 1–8. doi: 10.1016/j.jtbi.2017.12.025

Zhang, L., and Kong, L. (2018b). iRSpot-PDI: identification of recombination spots by incorporating dinucleotide property diversity information into Chou's pseudo components. *Genomics* 111, 457–464. doi: 10.1016/j.ygeno.2018.03.003

Zhang, Q., Zhu, L., and Huang, D. S. (2019). High-order convolutional neural network architecture for predicting DNA-protein binding sites. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 16, 1184–1192. doi: 10.1109/TCBB.2018.2819660

Zhang, X., Lu, X., Shi, Q., Xu, X. Q., Leung, H. C. E., Harris, L. N., et al. (2006). Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics* 7:197. doi: 10.1186/1471-2105-7-197

Zhou, T., Weng, J., Sun, X., and Lu, Z. (2006). Support vector machine for classification of meiotic recombination hotspots and coldspots in *Saccharomyces cerevisiae* based on codon composition. *BMC Bioinformatics* 7:223. doi: 10.1186/1471-2105-7-1

Zhu, Z., Albadawy, E., Saha, A., Zhang, J., Harowicz, M. R., and Mazurowski, M. A. (2019). Deep learning for identifying radiogenomic associations in breast cancer. *Comput. Biol. Med.* 109, 85–90. doi: 10.1016/j.compbiomed.2019.04.018

Zuo, Y. C., Su, W. X., Zhang, S. H., Wang, S. S., Wu, C. Y., Yang, L., et al. (2015). Discrimination of membrane transporter protein types using K-nearest neighbor method derived from the similarity distance of total diversity measure. *Mol. Biosyst.* 11, 950–957. doi: 10.1039/C4MB00681J

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.