Computational and Structural Biotechnology Journal 19 (2021) 4961–4969

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

EMCBOW-GPCR: A method for identifying G-protein coupled receptors based on word embedding and wordbooks

Wangren Qiu [a,*], Zhe Lv [a], Xuan Xiao [a,*], Shuai Shao [a], Hao Lin [b,*]

[a] School of Information Engineering, Jingdezhen Ceramic Institute, Jingdezhen, China
[b] Center for Informational Biology, University of Electronic Science and Technology of China, Chengdu 610054, China

ARTICLE INFO

Article history:
Received 27 April 2021
Received in revised form 7 August 2021
Accepted 27 August 2021
Available online 31 August 2021

Keywords:
Natural language processing
Word embedding
Bag-of-words
Extreme gradient boosting
Deep-learning
GPCRs

ABSTRACT

G Protein-Coupled Receptors (GPCRs) are one of the largest membrane protein receptor family in human, which are also important targets for many drugs. Thence, it's of great significance to judge whether a protein is a GPCR or not. However, identifying GPCRs by experimental methods is very expensive and time-consuming. As more and more GPCR primary sequences are accumulated, it's feasible to develop a computational model to predict GPCRs precisely and quickly. In this paper, a novel method called EMCBOW-GPCR has been proposed to improve the accuracy of identifying GPCRs based on natural language processing (NLP). For representing GPCRs, three word-embedding models and a bag-of-words model are used to extract original features. Then, the original features are thrown into a Deep-learning algorithm to extract features further and reduce the dimension. Finally, the obtained features are fed into Extreme Gradient Boosting. As shown with the results comparison, the overall prediction metrics of EMCBOW-GPCR are higher than the state of the arts. In order to be convenient for more researchers to use EMCBOW-GPCR, the method and source code have been opened in github, which are available at https://github.com/454170054/EMCBOW-GPCR, and a user-friendly web-server for EMCBOW-GPCR has been established at http://www.jci-bioinfo.cn/emcbowgpcr.

1. Introduction

G protein coupled receptors (GPCRs) are one of the largest family of membrane proteins in mammalian genomes which are widely distributed among the central nervous system, immune system, cardiovascular system, retina and other organs and tissues [1–4]. GPCRs can be divided into six classes [5]: rhodopsin-like receptors, secretin-like receptors, metabo-tropic glutamate receptors, fungal mating pheromone receptors, cyclic AMP receptors and frizzled receptors. What's more, they can regulate a wide range of physiological processes such as neurotransmission, growth, immune responses and so on [6–9]. Due to their structural characteristics and key role in signal transduction, GPCRs are the most important drugs target in modern drug research and development [10,11]. Therefore, identifying GPCRs accurately is very significant for drug development.

As more and more public GPCRs data is available, there are many efficient methods based on extracting features from sequence that are proposed to predict GPCRs in recent years. These methods usually consist of two parts: classification algorithm and feature extraction. The classification algorithms mainly based on statistics and machine learning methods, including Artificial Neural Network (ANN) [12,13], Random Forest(RF) [5,14], intimate sorting [15], K-Nearest Neighbor(KNN) [16,17], etc. The methods of feature representation for predicting GPCRs contain amino acid composition (AAC) [16,18], 400D [5], N-gram [5,13,19], SVM-Prot [14], etc. Zou [14] proposed a novel method in which the GPCRs were represented by a 188D feature vectors of SVM-Prot and the synthetic minority oversampling technique (SMOTE) [20–22] algorithm was used to generate some new positive samples to balance the training datasets. Finally, the prediction method adopted RF algorithm to be trained with the datasets. Recently, Yu [5] used the mixed-feature extraction methods to acquire the feature vector of GPCRs. In the work, three feature engineering methods including 400D, N-Gram and parallel correlation pseudo amino acid composition (PC-PseAAC) were chosen to extract features of GPCRs, respectively. Subsequently, these three feature vectors are randomly arranged and combined to form the mixed-feature. Further, the max relevance max distance (MRMD) [23] was employed to reduce the dimension of mixed-feature. According to the result,

* Corresponding authors.
E-mail addresses: qiuone@163.com (W. Qiu), jdzxiaoxuan@163.com (X. Xiao), hlin@uestc.edu.cn (H. Lin).

https://doi.org/10.1016/j.csbj.2021.08.044
2001-0370/

the mixed-feature concatenated by 400D and PC-PseAAC can achieve the best performance with RF algorithm. Although these methods have achieved positive results on predicting GPCRs, there is still room for improvement.

As a new research hotspot in artificial intelligence, Deep learning (DL) [24–28] is more and more widely used in machine learning. Because of its great help to the interpretation of data such as text, image and sound, DL has achieved many positive results in speech machine translation and image recognition far beyond previous related technologies, and has been successfully applied in many fields such as bioinformatics [29,30], computer vision [31,32], natural language processing [33,34], Automatic driving [35,36]and so on. In this work, we propose a novel method called EMCBOW-GPCR to predict GPCRs based on word embedding [37,38], BOW [39] and DL models. Firstly, we split the GPCRs sequences into segments of different lengths, and train the corresponding word embedding model with the split segments. Further, the every GPCR sequence is inputted into the word embedding models to get the word vectors. A BOW model was used to extract features at the same time. Secondly, the features by extracting from different methods are concatenated to form the original feature vectors. Thirdly, the original feature vectors are fed into a DL model to reduce the dimension and extract features further. Finally, the processed features are thrown into XGBoost algorithm to train a predictor. According to the results compared with other methods tested with the same data and performance measurement, our method can have a better performance.

2. Datasets and methods

2.1. Experimental datasets and performance measurement

The benchmark dataset used for evaluating the proposed method is the same as that used in literatures [5,14] and is available at https://github.com/454170054/EMCBOW-GPCR/blob/-main/files. The dataset sequences were download from UniProt [40] database and CD-Hit [41,42] program was used to reduce the sequence homology [14]. The sequence identity threshold was 0.8. The evaluation indicators used to test the performance of the methods in the work are Accuracy (Acc), Precision (Pre), Sensitivity (Sn), Specificity (Sp) and Matthews correlation coefficient (MCC) [43–45], which are listed in formula (1) explicitly.

$$\begin{cases} Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \\ Precision = \frac{TP}{TP+FP} \\ Sensitivity = \frac{TP}{TP+FN} \\ Specificity = \frac{TN}{TN+FP} \\ Strength = \frac{Sensitivity+Specificity}{2} \\ MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)+(TN+FP)(TN+FN)}} \end{cases} \quad (1)$$

where *TP* is the number of sequences that are GPCRs in fact and predicted as GPCRs,*TN* is the number of sequences that are non-GPCRs in fact and predicted as non-GPCRs, *FP*is the number of sequences that are non-GPCRs predicted as GPCRs,*FN*is the number of sequences that are GPCRs predicted as non-GPCRs. Further, we also apply Area Under ROC Curve (AUC) metric to evaluate the methods.

2.2. Feature extraction methods

In this study, two technologies are applied to extract features from GPCRs respectively. One of them is BOW [39,46] model which has been confirmed to be powerful in extracting GPCRs features based on sequences, and the other is Word Embedding which is very popular and important concept in natural language processing (NLP). The detailed process of the two feature extraction methods is listed as follows:

2.3. Word embedding

Word Embedding is a method of converting words into number vectors. The process of word embedding is to embed a high-dimensional space with all the number of words into a much lower dimensional and continuous vector space and each word or phrase is mapped to a vector in the real number field, and the result of word embedding generates a word vector which is the important technology of the task. In this paper, we trained three kinds of word embedding models and utilized them to generate corresponding feature vectors. The explicit training process is showed as follows:

2.3.1. Step 1: Splitting GPCRs sequences into fragments and create wordbooks

In order to satisfy the input data shape of the three word-embedding models, the GPCRs sequences are broken into different length fragments which are considered as words in wordbooks. In this paper, we designed three kinds fragments, and the length l of them can be set as 2, 3 or 4, respectively, and the wordbooks were denoted as $Q_{l=2}$, $Q_{l=3}$ and $Q_{l=4}$. For example, in order to obtain $Q_{l=2}$, the original sequences are broken into words whose lengths are 2, i.e., the window size is set 2 and the stride of moving window is 1. After all of this work, the words broken from each sequence would be collected, removed duplicate(s) and then form wordbook $Q_{l=2}$ of which the number of words is v. For example, the process of splitting GPCRs sequences into words of $Q_{l=2}$ is shown in Fig. 1. The processes of creating $Q_{l=3}$ and $Q_{l=4}$ are similar to $Q_{l=2}$. In detail, the window sizes of $Q_{l=3}$ and $Q_{l=4}$ should be set as 3 and 4, respectively. What's more, the strides of moving window of $Q_{l=3}$ and $Q_{l=4}$ are equal to 1.

2.3.2. Step 2: Training CBOW models

There are a lot of methods in word embedding. Here, the Word2vec [37] method is selected as the default word embedding method in this paper for the reason that Word2vec whose models are simple. Actually, a double layer neural networks have two widely used models to generate word vectors including continuous bag of words (CBOW) and Skip-gram. CBOW is applied to predict target words based on consecutive words before and after target word. Conversely, Skip-gram is applied to predict context words based on a word. In this work, CBOW model is chosen as the default model for word embedding. The structure of CBOW is shown in Fig. 2. The training of artificial neural network (ANN) [19,47,48] usually includes two parts: forward propagation and back propagation. The forward propagation calculation of the proposed model is listed as follows:

1). Encoding the GPCRs primary sequence with characters string. Since the original GPCRs sequences can not be directly fed into the CBOW model, a GPCR sequence can be represented with formula (2), where L is the length of the protein sequence.

$$G = g_1 g_2 g_3 g_4 \cdots g_L \quad (2)$$

2). Partitioning the sequence into word set.

$$R(B) = \{g_1 \cdots g_l, g_2 \cdots g_{l+1}, g_{L-l+1} \cdots g_L\} = \{R_1, R_2, \cdots, R_B\} \quad (3)$$

where B means the number of words in the set. Obviously, Bequals

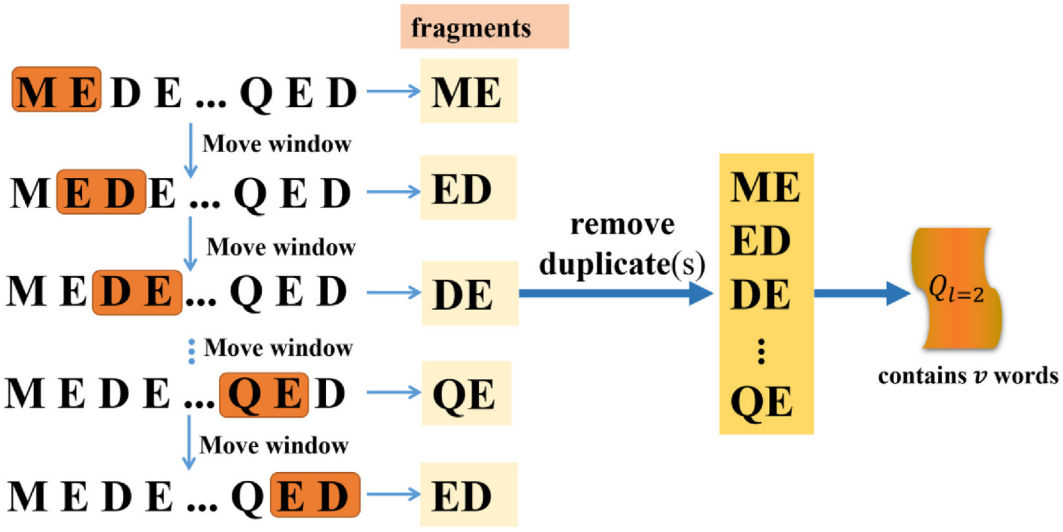

Fig. 1. The process of splitting GPCRs sequences and forming wordbook of $Q_{l=2}$.

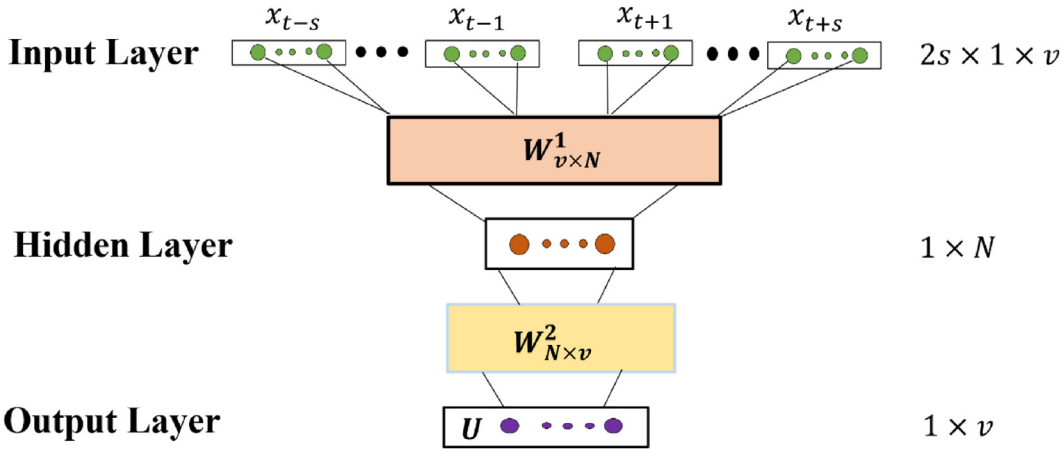

Fig. 2. The structure of CBOW.

$L - l + 1$, and l is the length of fragments.

3). Inputting the encoded words to CBOW and calculating the output of the hidden layer. Select the target word from $R(B)$, mark it as w_t and choose its context $w_{t-s}, \cdots, w_{t-1}, w_{t+1}, w_{t+s}$, i.e. s words from the upstream and s words from downstream of the target word in the protein sequences respectively. According to the corresponding wordbook created in **Step 1,** encoding the selected context words by using One-Hot and mark as $x_{t-s}, \cdots, x_{t-1}, x_{t+1}, \cdots, x_{t+s}$. Then feed them as the input data into Input Layer of CBOW. Calculation process from input layer to hidden layer is shown in formula (4).

$$h = \frac{1}{2s} \sum_{\substack{i=-s \\ i \neq 0}}^{s} x_{t+i} * W^1 \tag{4}$$

where W^1 is the weight matrix between Input Layer and Hidden Layer, whose shape is $v \times N$. N is a hyperparameter.h is the output of Hidden Layer, whose shape is $1 \times N$.

4). Designing the object function based on the propagation process from Hidden Layer to Output Layer. The formula (5) was utilized to compute the score matrix for the wordbook.

$$U = h * W^2 = (u_1, u_2, \cdots, u_j, \cdots, u_v) \tag{5}$$

where W^2 is the weight matrix between Hidden Layer and Output Layer, whose shape is $N \times v$. Then *Softmax* function is used to obtain the probability distribution of output units.

$$p(w_j) = \frac{e^{u_j}}{\sum_{i=1}^{v} e^{u_i}} \tag{6}$$

where u_j is the jth value of U and w_j means the jth word in wordbook.

To maximize the probability of the target word w_t, formula (7) was used at this step.

$$\max(p(w_t) = \frac{e^{u_t}}{\sum_{i=1}^{v} e^{u_i}}) \tag{7}$$

Take logarithm of formula (7) to get the objective function, and then maximize the objective function with formula (8).

$$loss = u_t - log \sum_{i=1}^{v} e^{u_i} \tag{8}$$

However, the above process usually needs a lot of time to train the model. Here, we adopt a more efficient technology called Negative Sampling [49] to accelerate the training process. By using

Negative Sampling can convert the above optimization problem to a series of binary problems and speed up training better word vectors. The detail of Negative Sampling and back propagation algorithm please see to reference [50,51]. In this step, we choose the value of N as 128, 256 and 512 on the basis of the number of words in relevant wordbook.

2.3.3. Step 3: Extracting features with CBOW models

In **Step 2,** we got three kinds of word vector matrixes W^1 by training CBOW models. In this step, each sequence was converted into feature vector. The detailed process is as follows:

1). Utilize formula (2) and (3) to process the GPCR sequence and get $R(B) = \{R_1, R_2, \cdots, R_B\}$.
2). According to the relevant wordbook, encode $R(B)$ by using One-Hot and then we can get $RO = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_b \end{bmatrix}$. Where r_i is a $1 \times v$ vector which represents the word R_i (see to formula (3)), $i = 1, 2, 3, \cdots, B$, and thus RO is a $B \times v$ matrix.
3). Calculate the feature vector matrixes by using formula (9)

$$MEF(l) = RO * W^1 \tag{9}$$

where $MEF(l)$ means matrix embedding features and is a $B \times N$ matrix.

4). Represent the GPCR sequence with formula (10).

$$EF(l) = \frac{1}{B} sum(MEF(l)) \tag{10}$$

where *sum* means to sum the values along the first dimension of the matrix and $EF(l)$ means embedding features.

With the same method, we can obtain $EF(2)$, $EF(3)$ and $EF(4)$, i.e. let l equals to 2, 3 or 4 for above formulas. And then the word embedding features (*WEF*) of each GPCR sequence can be combined into an 896-D vector by using the formula (11).

$$WEF = EF(2) \otimes EF(3) \otimes EF(4) \tag{11}$$

where $\oplus$ means concatenating the two vectors.

2.3.3.1. Bow. The brief steps of BOW features extraction from GPCRs are listed as follows:

(1). Encoding GPCRs sequences by using AAindex [52]which is a database containing more than 500 amino acid indices. The symbol 'X' existed in some sequences are represent with the mean of AAindex.
(2). Splitting GPCRs sequences into fragments with different sizes.
(3). Creating wordbooks and determining the number of words in each wordbook with weighted Silhouette Coefficient.
(4). Extracting 115-D BOW features denoted as *BF* based on wordbooks.

The specific process of BOW features extraction can be found in reference [46].

Finally, the above feature vectors are concatenated into a 1011-D vector by using formula (12) denoted as *F_GPCR*.

$$F\_GPCR = WEF \otimes BF \tag{12}$$

2.4. Feature extraction by Deep learning

In this work, we built a simple DL model which contains three fully connected layers and two Batch Normalization (BN) [53] layers to reduce the dimensions of the feature vector *F_GPCR*. In fact, the structure of DL model is very flexible. In order to simplify the problem, the number of neurons of each hidden layer is about half less than the previous layer. The detailed structure is shown in Fig. 3. What's more, we choose Focal loss [32] as the default loss function. Compared with other loss functions, Focal loss can handle the imbalance problem of datasets better [32].In this paper, the DL model is built by Tensorflow which is a very popular machine learning package. The hyperparameters of the DL model including epochs, batch size, loss function and optimizer are 20 and 32, binary cross-entropy and Adam [54]. The activation function of hidden layers is chosen Leaky Relu [55]. And the initial learning rate is 0.01. What's more, we used Early Stopping method to decide when to stop training. The strategy of Early Stopping could monitor the training loss and would stop the training process if the training loss did not decrease in the next 3 epochs. The model would be trained with the training dataset. Then the 1011-D features vector *F_GPCR* would be input into the optimized model and the output of Layer 1 would be intercepted as the final features. From Fig. 3, we can see that the final features are 505-D obviously after using DL model to reduce dimension.

2.5. Algorithm selection

2.5.1. Gradient boosting decision tree

Gradient Boosting Decision Tree (GBDT) is an effective machine learning algorithm in industry [56] and scientific research [46,57]. In practical research and application, classification and Regression Trees (CART) [58] is usually served as weak classifier for GBDT, and GBDT algorithm is trained through multiple iterations, each iteration produces a weak classifier which is trained on the basis of the residual of the previous round. The final total classifier (GBDT) is the weighted sum of the weak classifiers from each iteration of training.

2.5.2. Random forest

Random Forest (RF) [59] is a classical ensemble algorithm in machine learning. Because of its flexibility and generalization ability, this algorithm has been applied in many fields, such as bioinformatics, data mining and marketing management. The base learners of RF are usually CART. When a new sample is needed to be classified, each decision tree in the forest will be judged and classified separately. RF depends on a vote of their predictions to decide the final classification result.

2.5.3. CatBoost

CatBoost [60] is a kind of boosting algorithm based on symmetric Trees, which is universal and can be applied to a wide range of fields and various problems. Compared with other machine learning algorithms, the algorithm has three advantages. First of all, it can automatically handle categorical features. Further, CatBoost uses combined category features to make use of the relationship between features, which greatly enriches the feature dimension. Last but not least, the time of model training and predicting is very short.

2.5.4. XGBoost

XGBoost [61] is a well-known boosting algorithm in machine learning, which is mainly used to solve supervised learning problems. It can handle many tasks such as regression, classification and sorting and is widely used in machine learning competitions, and has achieved good results. XGBoost is generally regarded as an improvement of GBDT algorithm and can be more flexible and efficient [62,63]. The base learner of XGBoost is CART.

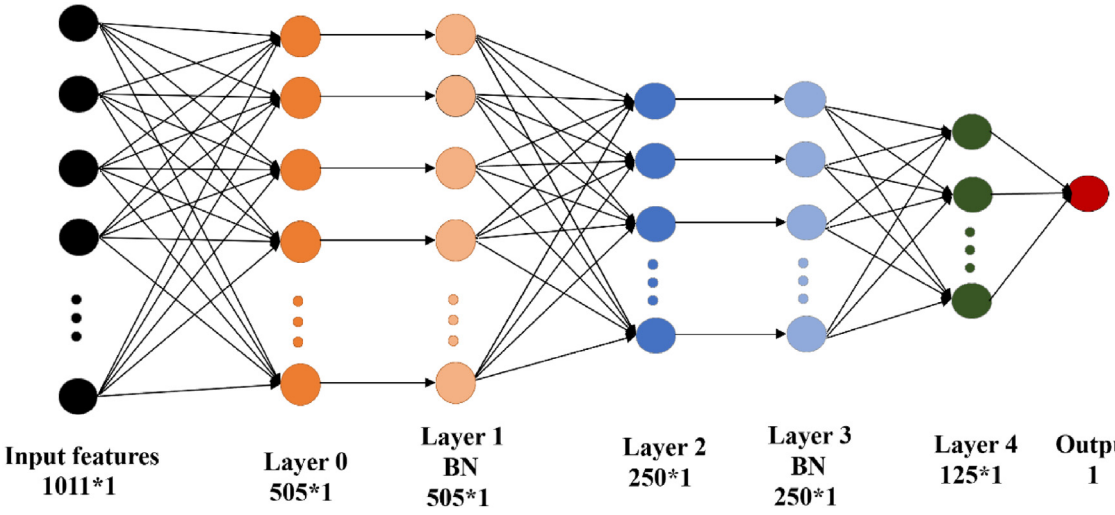

Fig. 3. The structure of deep learning model.

3. Results

3.1. Train the better CBOW models

From the mathematical derivation of CBOW, it is clear that getting a best CBOW model through training is to maximize the objective function. Here, a large value of iterations was set for the training of CBOW models and the best iterations were chosen by observing the change of objective function in the training process. In this paper, the all CBOW models were built and trained by using the software package of Python called Gensim. The code of training CBOW models is shown at https://github.com/454170054/EMCBOW-GPCR/blob/main/code/GetCBO

WFeatures.py, which contains the values of hyperparameters. Increment curves of objective functions of the three built models are shown in Fig. 4. From the picture, we can see when the objective functions of the models start to converge. From the left one in the picture, it is clear that when iterations of $Q_{l=2}$ is bigger than 74, the increment of objective function is close to 0. Therefore, we choose the result after the 74th training as the default CBOW model of $Q_{l=2}$. Similarly, the default CBOW models of $Q_{l=3}$ and $Q_{l=3}$ are the result after the 100th training and 280th training, respectively.

3.2. Effect of different feature representations of GPCR

In this work, we used two feature extraction methods, i.e., BOW features and Word Embedding features, to represent GPCRs sequences. In this section, the performances are compared based on mentioned-above single kind of features and combined features which are generated by concatenating BOW features and Word Embedding features by 5-fold cross-validation with 20 times on benchmark dataset. In this section, CatBoost algorithm are selected as the default algorithm to build classifiers. The experimental results about AUC values of different features are shown in Fig. 5. From the figure, we can see that the combined features whose AUC is 0.9423 which is better than CBOW features (AUC = 0.9391) and BOW features (AUC = 0.9344). According to the results, the combined features are selected as the default features to represent GPCRs sequences.

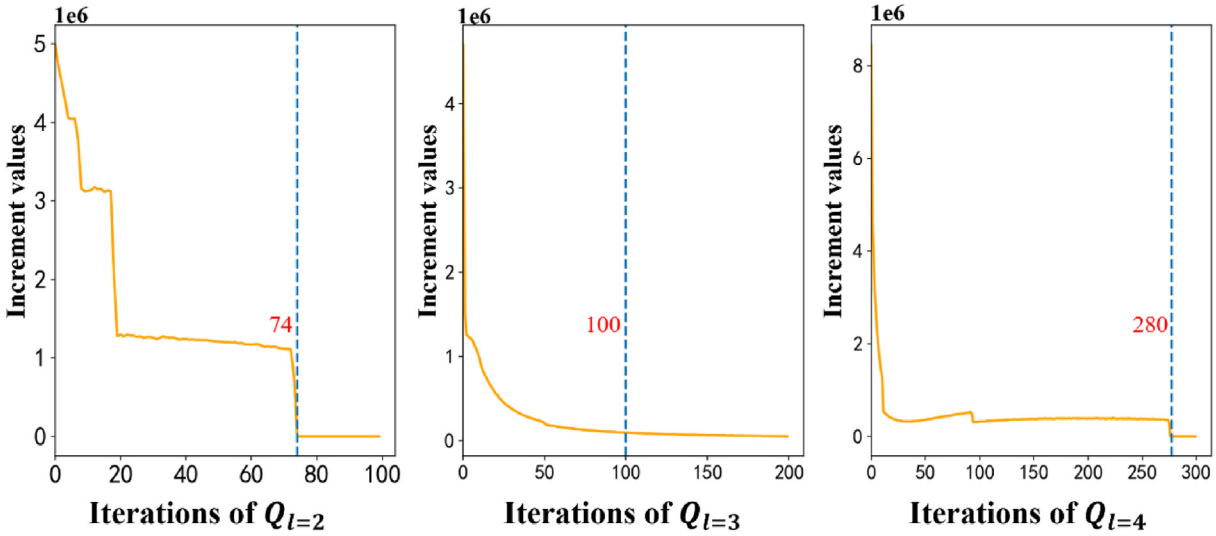

Fig. 4. Increment curves of objective functions of training CBOW.

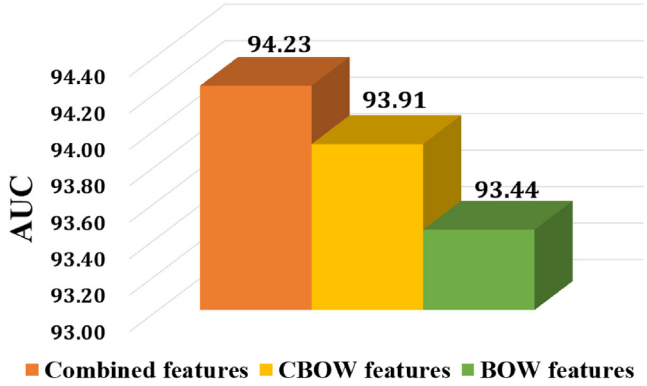

Fig. 5. AUC values of CatBoost algorithm combined with different features.

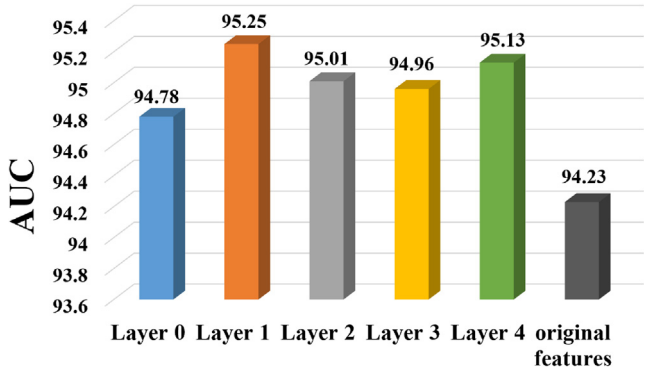

Fig. 6. AUC values of CatBoost algorithm combined with features generated by different layers.

3.3. Effect of different features generated by Deep-learning model

In this paper, a simple fully connected DL model was built with one input layer, three hidden layers, two BN layers and one output layer. Because of the model having three hidden layers and two BN layers, there are 5 kinds of features generated by the DL model. In this section, we evaluate the effectiveness of above features by 5-fold cross-validation with 20 times on benchmark dataset and the results are shown in Fig. 6. The details about the DL model are exhibited at https://github.com/454170054/EMCBOW-GPCR/blob/main/code/GetResults.py. CatBoost is chosen as the default algorithm to build predictive model to get AUC values in this section. From the figure, we can find that the AUC value of original features is the smallest. This result can demonstrate the fact that the processed features by DL have better performance. What's more, the features generated by Layer 1 of DL model achieved the biggest value of AUC so that it was chosen as the default features.

3.4. Determination of the optimal algorithm

Up to now, there are many effective algorithms developed in the field of machine learning. Ensemble learning often performs better than the best single algorithm since it would construct a certain number of weak classifiers and then classifies a new sample by taking a (weighted) vote of their predictions. The all learning algorithms mentioned in section **3** belong to ensemble learning, in which the random forest is part of Bagging method and the rest is part of Boosting method [64]. In this section, the above four algorithms are tested on the benchmark dataset by 5-fold cross-validation with 20 times and the performance of them is listed in Table 1 and the input features used there is the features generated by Layer 1 of the DL model. As shown in the table, the performance of DL model is worse than other algorithms which are using the features generated by DL model. What's more, The RF algorithm

Table 4
The results of the proposed method and literature [5].

Method	AUC	Acc	Precision
Proposed method	**95.29**	**88.11**	**89.28**
400D + PC-PseAAC	94.13	86.28	86.62

Table 1
Performance of different algorithms.

Algorithm	Acc	Sp	Sn	MCC	AUC
CatBoost	92.87±0.14	97.06±0.13	**75.43±0.54**	76.29±0.48	95.25±0.14
RF	92.85±0.10	**98.40±0.07**	69.78±0.44	75.85±0.38	95.38±0.09
GBDT	92.61±0.13	96.77±0.12	75.27±0.52	75.48±0.58	94.82±0.13
XGBoost	**92.91±0.10**	97.34±0.11	74.45±0.41	**76.32±0.34**	**95.56±0.13**
DL	92.23±0.30	97.56±0.47	70.03±2.65	73.77±1.14	93.80±0.35

Notice: digits are mean± std and the bold means the best values.

Table 2
The hyperparameters used for the classifiers.

Algorithm	n_estimators	Learning rate	Max depth
CatBoost	190	0.4	None
RF	110		None
GBDT	100	0.1	3
XGBoost	110	0.12	3

Notice: slash means algorithm do not have the hyperparameter.

Table 3
The results of the proposed method and Liao [14].

Method	Acc	Sp	Sn	MCC	AUC
Proposed method	**92.91±0.10**	**97.34±0.11**	**74.45±0.41**	**76.32±0.34**	**95.56±0.13**
Liao	83.33±7.26	97.24±0.87	69.42±14.91	69.24±12.96	92.80±3.8

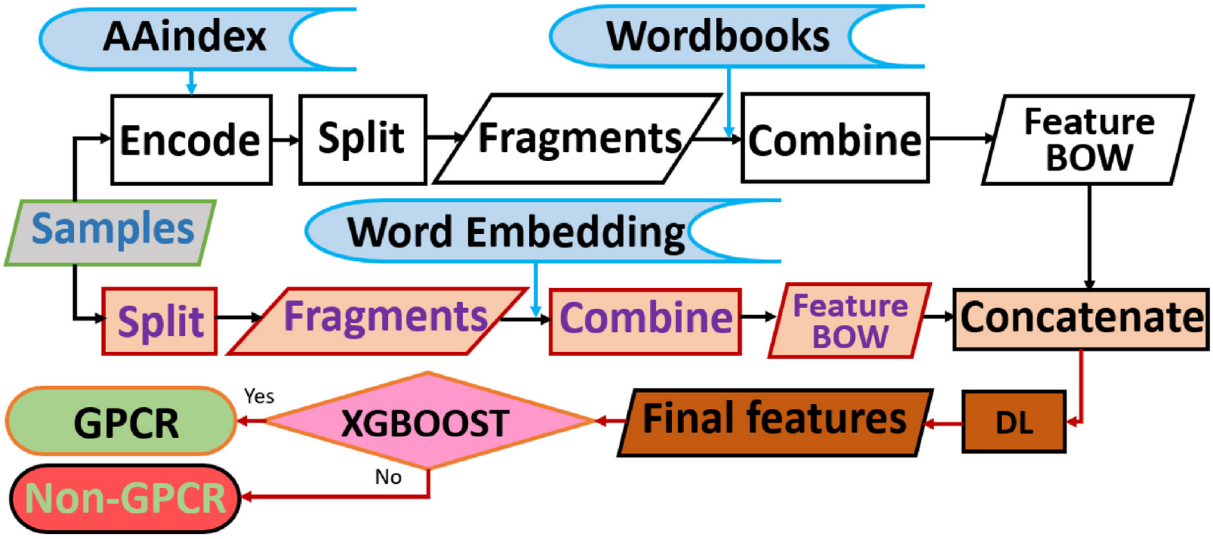

Fig. 7. The framework of proposed method EMCBOW-GPCR.

has the largest Sp but having the smallest Sn and CatBoost algorithm has the best Sn. Furthermore, the Acc, Mcc and AUC value of XGBoost is the highest. Therefore, XGBoost algorithm is selected as the final algorithm to build classifier in this work. The hyperparameters used for the classifiers are included in Table 2 and the software packages can be found in https://github.com/454170054/EMCBOW-GPCR/blob/main/requirements.txt.

3.5. Comparison of other methods

In this work, the benchmark dataset is same as the one used in [5,14]. In order to prove the effectiveness of our method, the performance is compared between the proposed method and the other state-of-the-art method. Because the data segmentation methods in the literature are different, the same segmentation strategy was test here. In the literature [14], the state-of-the-art method was tested on the benchmark dataset by 5-fold cross-validation. Further, SMOTE algorithm was used to balance the training dataset and change the positive samples from 100 percent into 300 percent. We also take the strategy to handle the benchmark dataset and test our proposed method. The results are shown in Table 3. It is clear that all the metrics of the proposed method are better than those of Liao. In the literature [5], the negative samples in the benchmark dataset were randomly divided into 4 groups and 2495 sequences were extracted from these 4 groups, respectively. Then, the final result is an average of the four experiment by using the four negative experiments. According to the strategy, we test our proposed method and the result is shown in Table 4. From the table, we can find that the AUC, Acc and Precision value of our method are higher than the other. The results of comparing with other methods show that the proposed method is a good method for identifying GPCRs.

4. Conclusion

In this work, CatBoost was chosen as the initial algorithm to build classifier for the reason that the time of training CatBoost is much shorter than other mentioned algorithms. Firstly, we selected the best CBOW models by optimizing the objective function. Secondly, BOW and CBOW models were employed to extract features separately and the metrics listed in formula (1) were used to evaluate the effectiveness of different feature representations for GPCRs. According to the experiment results, the combined features by concatenating BOW and Word Embedding features were better than any single features. Therefore, we determined to choose the combined features as the default features to represent GPCRs sequences. Then, a simple DL model was built and CatBoost was employed to fit the processed features generated by different layers of DL model and generate corresponding classifier. According to the performance of different classifiers, the outputs of Layer 1 were chosen as the final features. Further, XGBoost was selected as the default algorithm because of its best performance compared with other algorithms. Finally, according to results compared with other state-of-the-art methods, the proposed method called EMCBOW-GPCR got a better performance in the problem of identifying GPCRs.

5. Discussion

G protein coupled receptors (GPCRs) family is one of the largest membrane protein family in human beings, and is also an important target of many drugs. In this work, a novel method for identifying GPCRs was developed. In terms of representation GPCRs, we used two extraction methods which are BOW and Word Embedding to extract features from GPCRs sequences. Then we concatenated the above two kinds of features as the input of DL which can automatically extract better features by learning from the input features. Further, we intercepted the output of Layer 1 as the input features of XGBoost which is a fairly powerful, flexible and efficient algorithm. According to the results of compared with other methods, the proposed method called EMCBOW-GPCR has a better performance for identifying GPCRs. By the way, the hyperparameter N which is used in Word Embedding is flexible and hard to determine the best value. What's more, it may be influenced the quality of features significantly. At the same time, the DL model is also very flexible and difficult to find the best structure which contains how many layers and units in each layer. Finally, the framework of EMCBOW-GPCR can be concluded as Fig. 7.

CRediT authorship contribution statement

Wangren Qiu: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Software, Validation, Writing - review & editing. **Zhe Lv:** Visualization, Investigation, Software, Validation. **Xuan Xiao:** Conceptualization, Methodology, Software,

Supervision, Writing - review & editing. **Shuai Shao:** Visualization, Investigation. **Hao Lin:** Conceptualization, Methodology, Software, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the grants from the National Natural Science Foundation of China (No. 31760315, 31860312), Natural Science Foundation of Jiangxi Province, China (NO. 20202BAB202007).

Author contributions

W. Q. conceived and designed the experiments; Z. L. and S. S. performed the extraction of features, model construction, model training, and evaluation. W.Q. and Z. L. analyzed the data and implemented the classifiers. Z. L. drafted the manuscript. X. X. and H. L. supervised this project and revised the manuscript. All authors read and approved the final manuscript.

References

[1] Lagerstrm MC, Schith HB. Lagerstrom, M. C. & Schioth, H. B. Structural diversity of G protein-coupled receptors and significance for drug discovery. Nature Rev. Drug Discov. 7, 339-357. Nature Reviews Drug Discovery 2008;7:339–57.
[2] Jacoby E, Bouhelal R, Gerspacher M, Seuwen K. The 7 TM G-protein-coupled receptor target family. ChemMedChem 2006;1:760–82. https://doi.org/10.1002/cmdc.200600134.
[3] Fredriksson R, Lagerström MC, Lundin L-G, Schiöth HB. The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints. Mol Pharmacol 2003;63(6):1256–72. https://doi.org/10.1124/mol.63.6.1256.
[4] Ramesh M, Soliman ME. G-protein coupled receptors (GPCRs): a comprehensive computational perspective. Comb Chem High Throughput Screen 2015;18:346–64. https://doi.org/10.2174/1386207318666150305155545.
[5] Ao C, Gao L, Yu L. Identifying G-protein Coupled Receptors Using Mixed-Feature Extraction Methods and Machine Learning Methods. IEEE Access 2020; PP:1.
[6] Eo H-S, Choi JP, Noh S-J, Hur C-G, Kim W. A combined approach for the classification of G protein-coupled receptors and its application to detect GPCR splice variants. Comput Biol Chem 2007;31(4):246–56.
[7] Baldwin JM. Structure and function of receptors coupled to G proteins. Curr Opin Cell Biol 1994;6(2):180–90.
[8] Chou K-C, Elrod DW. Bioinformatical analysis of G-protein-coupled receptors. J Proteome Res 2002;1(5):429–33.
[9] Katritch V, Cherezov V, Stevens RC. Structure-function of the g protein-coupled receptor superfamily. Annu Rev Pharmacol Toxicol 2013;53(1):531–56.
[10] Zhang Ru, Xie X. Tools for GPCR drug discovery. Acta Pharmacol Sin 2012;33 (3):372–84. https://doi.org/10.1038/aps.2011.173.
[11] Alexander SP, Mathie A, Peters JA. Guide to receptors and channels (GRAC). Br J Pharmacol 2011;164(Suppl 1):S1–324. https://doi.org/10.1111/j.1476-5381.2011.01649_1.x.
[12] Zia Ur R, Khan A. Identifying GPCRs and their types with Chou's pseudo amino acid composition: an approach from multi-scale energy representation and position specific scoring matrix. Protein Pept Lett 2012;19:890–903. https://doi.org/10.2174/092986612801619589.
[13] Li M, Ling C, Xu Qi, Gao J. Classification of G-protein coupled receptors based on a rich generation of convolutional neural network, N-gram transformation and multiple sequence alignments. Amino Acids 2018;50(2):255–66. https://doi.org/10.1007/s00726-017-2512-4.
[14] Liao Z, Ju Y, Zou Q. Prediction of G protein-coupled receptors with SVM-prot features and random forest. Scientifica 2016;2016:1–10.
[15] Peng Z-L, Yang J-Y, Chen X. An improved classification of G-protein-coupled receptors using sequence-derived features. BMC Bioinf 2010;11(1):420. https://doi.org/10.1186/1471-2105-11-420.
[16] Naveed M, Khan AU. GPCR-MPredictor: multi-level prediction of G protein-coupled receptors using genetic ensemble. Amino Acids 2012;42:1825.
[17] Dongardive J, Abraham S, Behera HS, Mohapatra DP. Protein sequence classification based on N-gram and K-nearest neighbor algorithm. New Delhi: Springer India; 2016. p. 163–71.
[18] Nie G, Li Y, Wang F, Wang S, Hu X, Liu F, et al. A novel fractal approach for predicting G-protein-coupled receptors and their subfamilies with support vector machines. Biomed Mater Eng 2015;26(s1):S1829–36. https://doi.org/10.3233/BME-151485.
[19] Li M, Ling C, Gao J. An efficient CNN-based classification on G-protein Coupled Receptors using TF-IDF and N-gram. 2017. doi: 10.1109/ISCC.2017.8024644.
[20] Wang X, Yu B, Ma A, Chen C, Liu B, Ma Q. Protein-protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. Bioinformatics 2019;35:2395–402. Doi: 10.1093/bioinformatics/bty995.
[21] Blagus R, Lusa L. SMOTE for high-dimensional class-imbalanced data. BMC Bioinf 2013;14:1–16.
[22] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Synthetic minority oversampling technique. J Artific Intell Res 2002;16:321–57.
[23] Zou Q, Zeng J, Cao L, Ji R. A novel features ranking metric with application to scalable visual and bioinformatics data classification. Neurocomputing 2016;173:346–54.
[24] Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw 2015;61:85–117.
[25] Hinton GE. Learning multiple layers of representation. Trends Cogn Sci 2007;11(10):428–34.
[26] Hao X, Zhang G, Ma S. Deep learning. Int J Seman Comput 2016;10 (03):417–39.
[27] Lv H, Dao F-Y, Guan Z-X, Yang H, Li Y-W, Lin H. Deep-Kcr: accurate detection of lysine crotonylation sites using deep learning method. Brief Bioinform 2020. 10.1093/bib/bbaa255.
[28] Wang D, Zhang Z, Jiang Y, Mao Z, Wang D, Lin H, et al. DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism. Nucleic Acids Res 2021. 10.1093/nar/gkab016.
[29] Duolin, Wang, Yanchun, Liang, Dong. Capsule network for protein post-translational modification site prediction. Bioinformatics 2019;35:2386–94.
[30] Li Y, Huang C, Ding L, Li Z, Pan Y, Gao X. Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. Methods 2019;166:4–21. https://doi.org/10.1016/j.ymeth.2019.04.008.
[31] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. IEEE Conf Comput Vis Pattern Recogn (CVPR) 2016;2016:770–8. https://doi.org/10.1109/CVPR.2016.90.
[32] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis & Machine Intelligence 2017; PP:2999–3007.
[33] Lin YO, Lei H, Li XY, Wu J. Deep Learning in NLP: Methods and Applications. Dianzi Keji Daxue Xuebao/Journal of the University of Electronic Science and Technology of China 2017;46:913–9.
[34] Chen G, Ye D, Xing Z, Chen J, Cambria E. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. Int Joint Conf Neural Netw (IJCNN) 2017;2017:2377–83. https://doi.org/10.1109/IJCNN.2017.7966144.
[35] Uçar A, Demir Y, Güzeliş C. Object recognition and detection with deep learning for autonomous driving applications. Simulation 2017;93(9):759–69. https://doi.org/10.1177/0037549717709932.
[36] Chen C, Seff A, Kornhauser A, Xiao J. DeepDriving: learning affordance for direct perception in autonomous driving. IEEE Int Conf Comput Vision (ICCV) 2015;2015:2722–30. https://doi.org/10.1109/ICCV.2015.312.
[37] Mikolov T, Corrado G, Kai C, Dean J. Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), 2013.
[38] Yang KK, Wu Z, Bedbrook CN, Arnold FH. Learned protein embeddings for machine learning. Bioinformatics 2018;34:4138. https://doi.org/10.1093/bioinformatics/bty455.
[39] Wang P, Huang X, Qiu W, Xiao X. Identifying GPCR-drug interaction based on wordbook learning from sequences. BMC Bioinf 2020;21:150. https://doi.org/10.1186/s12859-020-3488-8.
[40] Boutet E, Lieberherr D, Tognolli M, Schneider M, Bansal P, Bridge AJ, et al. UniProtKB/Swiss-prot, the manually annotated section of the UniProt KnowledgeBase: how to use The entry view. Methods Mol Biol 2016;1374:23–54. https://doi.org/10.1007/978-1-4939-3167-5_2.
[41] Li W, Godzik A. Cd-Hit: a fast program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences. Bioinformatics (Oxford, England) 2006;22:1658–9. https://doi.org/10.1093/bioinformatics/btl158.
[42] Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 2012:3150–2.
[43] Zhang D, Chen H-D, Zulfiqar H, Yuan S-S, Huang Q-L, Zhang Z-Y, et al. iBLP: an XGBoost-Based Predictor for Identifying Bioluminescent Proteins. Comput Math Methods Med 2021;2021:6664362. https://doi.org/10.1155/2021/6664362.
[44] Dao FY, Lv H, Zhang D, Zhang ZM, Liu L, Lin H. DeepYY1: a deep learning approach to identify YY1-mediated chromatin loops. Brief Bioinform 2020. https://doi.org/10.1093/bib/bbaa356.
[45] Chen W, Feng P, Nie F. iATP: a sequence based method for identifying anti-tubercular peptides. Med Chem 2019. https://doi.org/10.2174/1573406415666191002152441.
[46] Qiu W, Lv Z, Hong Y, Jia J, Xiao X. BOW-GBDT: a GBDT classifier combining with artificial neural network for identifying GPCR-drug interaction based on wordbook learning from sequences. Front Cell Dev Biol 2020;8:. https://doi.org/10.3389/fcell.2020.623858623858.

[47] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back propagating errors. Nature 1986;323:533–6.
[48] Judith EDPD, Deleo JM. Artificial neural networks. Cancer 2001;91:1615–35.
[49] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 2013:3111–9.
[50] Rong X. word2vec Parameter Learning Explained. Computer Science 2014.
[51] Bottou L, Lechevallier Y, Saporta G. Large-scale machine learning with stochastic gradient descent. Heidelberg: Physica-Verlag HD; 2010. p. 177–86.
[52] Kawashima S, Kanehisa M. AAindex: amino acid index database. Nucleic Acids Res 2000;28:374.
[53] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning, PMLR; 2015, p. 448–56.
[54] Kingma DP, Ba J. Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:14126980 2014.
[55] Maas AL, Hannun AY, Ng AY, others. Rectifier nonlinearities improve neural network acoustic models. Proc. icml, vol. 30, 2013, p. 3.
[56] He X, Pan J, Jin O, Xu T, Liu B, Xu T, et al. Practical Lessons from Predicting Clicks on Ads at Facebook. Proceedings of the Eighth International Workshop on Data Mining for Online Advertising 2014:1–9. 10.1145/2648584.2648589.
[57] Tian D, He G, Wu J, Chen H, Jiang Y. An accurate eye pupil localization approach based on adaptive gradient boosting decision tree. Vis Commun Image Process (VCIP) 2016;2016:1–4. https://doi.org/10.1109/VCIP.2016.7805483.
[58] Friedman. Classification and Regression Trees. Wadsworth International Group; 1984.
[59] Breiman L. Random forests. Mach Learn 2001;45:5–32.
[60] Bentéjac C, Csrg A, Martínez-Muoz G. A comparative analysis of gradient boosting algorithms. Artif Intell Rev 2020:1–31.
[61] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016:785–94. 10.1145/2939672.2939785.
[62] Yang H, Luo Y, Ren X, Wu M, He X, Peng B, et al. Risk prediction of diabetes: big data mining with fusion of multifarious physical examination indicators. Inform Fusion 2021. https://doi.org/10.1016/j.inffus.2021.02.015.
[63] Tang Q, Nie F, Kang J, Chen W. mRNALocater: enhance the prediction accuracy of eukaryotic mRNA subcellular localization by using model fusion strategy. Mol Ther 2021. https://doi.org/10.1016/j.ymthe.2021.04.004.
[64] Dietterich TG. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn 2000;40:139–57.