Interdisciplinary Sciences: Computational Life Sciences (2022) 14:209–232
https://doi.org/10.1007/s12539-021-00483-y

ORIGINAL RESEARCH ARTICLE

EnANNDeep: An Ensemble-based lncRNA–protein Interaction Prediction Framework with Adaptive *k*-Nearest Neighbor Classifier and Deep Models

Lihong Peng[1,2] · Jingwei Tan[1] · Xiongfei Tian[1] · Liqian Zhou[1]

Received: 6 July 2021 / Revised: 14 September 2021 / Accepted: 15 September 2021 / Published online: 10 January 2022

Abstract

lncRNA–protein interactions (LPIs) prediction can deepen the understanding of many important biological processes. Artificial intelligence methods have reported many possible LPIs. However, most computational techniques were evaluated mainly on one dataset, which may produce prediction bias. More importantly, they were validated only under cross validation on lncRNA–protein pairs, and did not consider the performance under cross validations on lncRNAs and proteins, thus fail to search related proteins/lncRNAs for a new lncRNA/protein. Under an ensemble learning framework (EnANNDeep) composed of adaptive *k*-nearest neighbor classifier and Deep models, this study focuses on systematically finding underlying linkages between lncRNAs and proteins. First, five LPI-related datasets are arranged. Second, multiple source features are integrated to depict an lncRNA–protein pair. Third, adaptive *k*-nearest neighbor classifier, deep neural network, and deep forest are designed to score unknown lncRNA–protein pairs, respectively. Finally, interaction probabilities from the three predictors are integrated based on a soft voting technique. In comparing to five classical LPI identification models (SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF) under fivefold cross validations on lncRNAs, proteins, and LPIs, EnANNDeep computes the best average AUCs of 0.8660, 0.8775, and 0.9166, respectively, and the best average AUPRs of 0.8545, 0.8595, and 0.9054, respectively, indicating its superior LPI prediction ability. Case study analyses indicate that SNHG10 may have dense linkage with Q15717. In the ensemble framework, adaptive *k*-nearest neighbor classifier can separately pick the most appropriate *k* for each query lncRNA–protein pair. More importantly, deep models including deep neural network and deep forest can effectively learn the representative features of lncRNAs and proteins.

✉ Lihong Peng
plhhnu@163.com

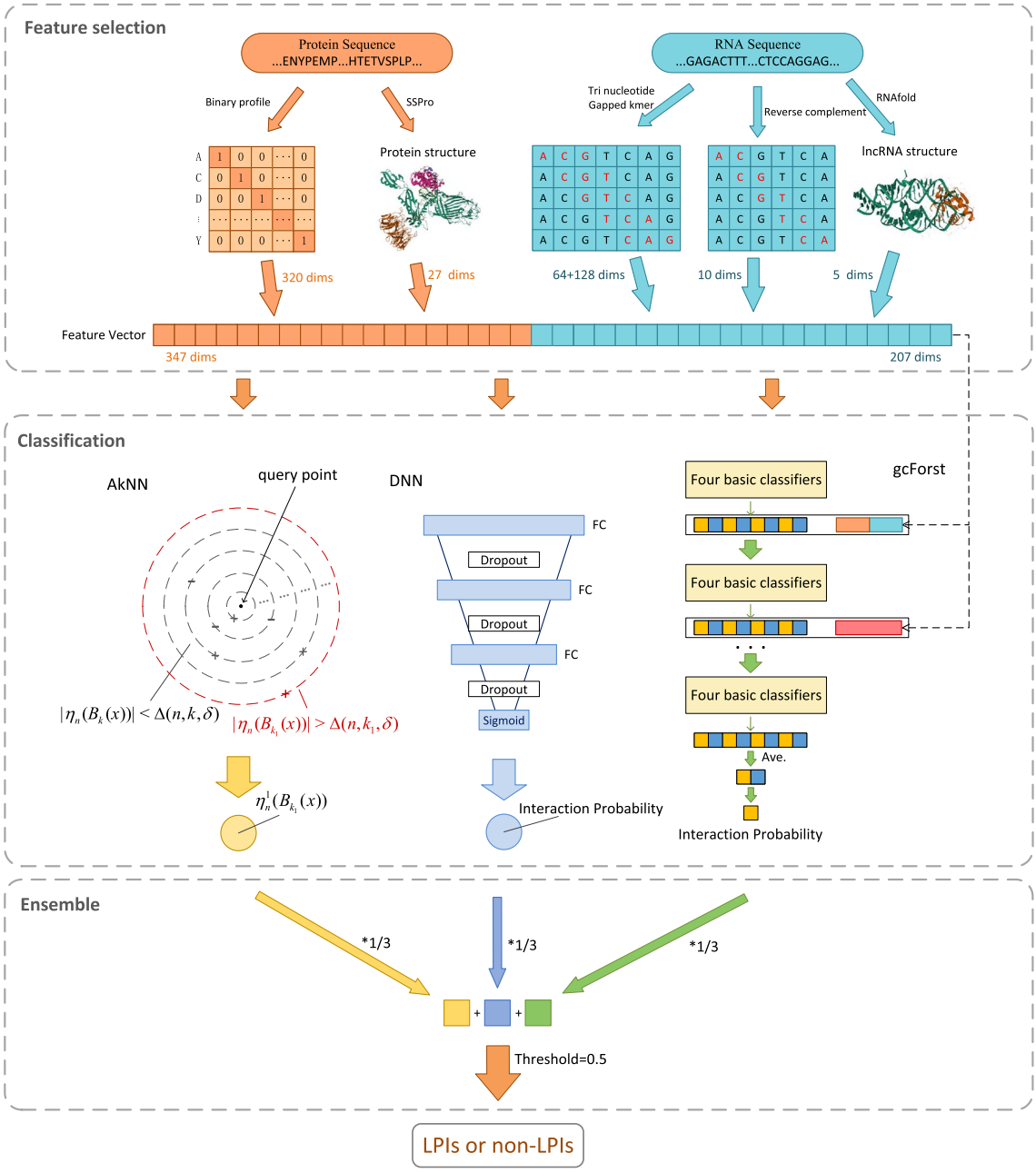
✉ Liqian Zhou
Zhoulq11@163.com

[1] School of Computer Science, Hunan University of Technology, Zhuzhou, China

[2] College of Life Sciences and Chemistry, Hunan University of Technology, Zhuzhou, China

Graphical abstract

EnANNDeep: An Ensemble-based lncRNA-protein Interaction Prediction Framework with Adaptive k-Nearest Neighbor Classifier and Deep Models



Keywords lncRNA–protein interaction · Adaptive *k*-nearest neighbor · Deep neural network · Deep forest · Ensemble learning

Abbreviations

LPI	Long noncoding RNA–Protein Interaction	*k*-NN	*k*-nearest neighbor
EnANNDeep	Ensemble-based lncRNA–protein interaction prediction framework with adaptive *k*-nearest neighbor and deep models	A*k*NN	Adaptive *k*-nearest neighbor
		DNN	Deep neural network
		CVs	Cross validations

1 Introduction

1.1 Motivation

Long noncoding RNAs (lncRNAs) are a class of long endogenous noncoding RNAs with poor sequence conservation [1–3]. lncRNAs have close association with multiple key biological processes [4]. More importantly, increasing works imply that lncRNAs also densely linking with many complex diseases [5, 6], for example, brachydactyly syndrome and HELLP syndrome [7], facioscapulohumeral muscular dystrophy [8], fat [9], and cancers. For example, lncRNAs HOXA-AS2 and SNHG12 are identified as possible therapeutic targets and biomarkers in human cancers [10, 11], DLEU1 densely links with colorectal cancer progression through the activation of KPNA3 [12], HOTAIR's expression is elevated in lung cancer [13], ZFAS1 has close relationship with cervical cancer cell chemosensitivity [14]. In summary, lncRNAs have been increasingly confirmed to be tumor-related biological molecules. However, to date, relationships between lncRNA and known tumor-suppressive entities remain largely elusive. Evidence indicates that lncRNAs exert their biological functions based on the linkages with RNA-binding proteins. Therefore, the identification of potential lncRNA–protein interactions (LPIs) contributes to understand many important biological processes and progression and metastasis of various complex diseases.

1.2 Related Work

Wet-lab experiments for LPI identification are time-consuming and waste of sources. Computational methods have been gradually explored for potential LPI discovery. Existing computation-based LPI prediction methods can be roughly categorized into network-based techniques and machine learning-based techniques. Network-based methods generally construct a few lncRNA/protein-related networks and then design a network algorithm to compute the probabilities of interactions between lncRNAs and proteins. Zhao et al. [15] and Ge et al. [16] designed two bipartite network-based recommended algorithm to score each lncRNA–protein pair. Zhou et al. [17] proposed a similarity kernel fusion method for LPI prediction (LPI-SKF). Zheng et al. [18] fused multiple protein similarity networks to uncover potential associations between lncRNAs and proteins.

Machine learning-based methods select features for lncRNAs and proteins to describe an lncRNA–protein pair, and use the extracted features as input to train a supervised learning model for possible LPI identification. The type of methods contain matrix factorization-based models, ensemble learning-based models, and deep learning-based models. To discover new LPIs, Liu et al. [19], Zhang et al. [20], and Ma et al. [21] explored neighborhood regularized logistic matrix factorization method, graph regularized nonnegative matrix factorization model, and projection-based neighborhood nonnegative matrix decomposition method (PMKDN), respectively.

Ensemble learning-based techniques have been widely available for LPI identification. Hu et al. [22] presented a unified framework combining support vector machines, random forests, and extreme gradient boosting. Zhang et al. [23] designed a feature projection ensemble learning-based framework (SFPEL). Deng et al. [24] picked lncRNA and protein information including HeteSim features and diffusion features and constructed a gradient tree boosting algorithm (PLIPCOM). Fan et al. [25] explored a broad learning system-based ensemble classification model. Wekesa et al. [26] exploited a categorical boosting approach (LPI-CatBoost). Yi et al. [27] proposed a stacking ensemble learning algorithm.

Deep learning architectures can better learn hidden information in raw data and characterize data in each layer based on nonlinear transformations [28]. Therefore, deep learning has been a research hotspot in the area of bioinformatics [6, 29–31]. In LPI prediction, deep learning demonstrates also broad application, such as the works provided by [32–35]. Deng et al. [32] proposed a deep neural network for predicting binding site of RNA-binding proteins. Wei et al. [35] fused biological feature blocks via Deep Neural Network (DNN). Zhang et al. [33] presented an ensemble deep learning model for identifying interaction biomolecule types for lncRNAs. Wekesa et al. [34] explored a graph attention-based deep learning model to predict plant LPIs. Zhao et al. [36] developed a graph convolutional network-based method to prioritize target protein-coding genes of lncRNAs. Shaw et al. [37] exploited a multimodal deep learning model to identify relationships between lncRNAs and protein isoforms.

Computational methods effectively discovered many potential relevances between lncRNAs and proteins. However, network-based techniques fail to find possible proteins/lncRNAs for an orphan lncRNA/protein. Machine learning-based LPI prediction approaches remain the following problems to solve. First, most methods are measured on one dataset, which may result in prediction bias. Second, the majority of methods are validated under Cross Validation (CV) on lncRNA–protein pairs, ignored the performance under the other CVs, for example, CVs on lncRNAs or proteins. Finally, features of lncRNAs and proteins are required to further integration. The details are summarized in Table 1.

1.3 Study Contributions

In this manuscript, an ensemble learning framework (EnANNDeep) is developed to quantify the interplays between lncRNAs and proteins. EnANNDeep integrates diverse biological information, Adaptive *k*-nearest neighbor (A*k*NN) classifier, deep neural network, Deep forest, and ensemble learning theory to a unified framework. The work mainly has the following three contributions:

Table 1 Summarization of existing studies and the proposed method

Method	Year	Model	Dataset	The type of CV
Network-based methods	2016	LPBNI	NPInter 2.0	CV_l
	2018	LPI-BNPRA	dataset 3	CV_{lp}
	2017	HeteSim algorithm	NPInter 2.0	CV_{lp}
	2020	LPI-SKF	dataset 3	CV_l, CV_p, and CV_{lp}
Machine learning-based methods	2017	LPI-NRLMF	dataset 3	CV_{lp}
	2018	PLIPCOM	NPInter 3.0	CV_{lp}
	2017	LPGNMF	NPInter 3.0	CV_{lp}
	2019	PMKDN	dataset 3	CV_l, CV_p, and CV_{lp}
	2018	SFPEL-LPI	dataset 3	CV_l, CV_p, and CV_{lp}
	2018	HLPI-Ensemble	NPInter 2.0	CV_{lp}
	2019	LPI-BLS	NPInter 3.0	CV_{lp}
Deep learning-based methods	2020	DRPLPI	datasets 4, 5	CV_{lp}
	2020	RPI-SE	NPInter 3.0	CV_{lp}
	2020	GPLPI	datasets 4 ,5	CV_{lp}
	2021	DeepLPI	NPInter 3.0	CV_{lp}

CV_l, CV_p, and CV_{lp} denote CV on lncRNAs, proteins, lncRNA–protein pairs

PLIPCOM Gradient Tree Boosting technique, *LPBNI* bipartite network, *LPI-BNPRA* bipartite network projection recommended algorithm, HeteSim algorithm, *LPI-SKF* similarity kernel fusion + Laplacian regularized least squares, *LPI-NRLMF* neighborhood regularization + logistic matrix factorization, *LPGNMF* graph regularization + nonnegative matrix factorization, *PMKDN* projection-based neighborhood + nonnegative matrix decomposition model, *SFPEL-LPI* feature projection + ensemble learning method, *HLPI-Ensemble* Support Vector Machines + Random Forests + Extreme Gradient Boosting, *LPI-BLS* Broad Learning System + stacked ensemble classifier with a logistical regression model, *DRPLPI* CatBoost + Extra Tree + LSTM Autoencoder, *RPI-SE* SVM + Gradient Boosting Decision Tree + Extremely randomized Trees algorithms, *GPLPI* Graph attention-based autoencoder + CatBoost and Logistic regression, *DeepLPI* Deep Neural Network + Conditional Random Field

1. An ensemble learning framework, composed of A*k*NN algorithm, DNN, and deep forest, is exploited to greatly learn labels of unknown lncRNA–protein pairs.
2. The proposed A*k*NN classification model separately selects the right k for each neighborhood and provides an upper bound for the failure probability.
3. Deep models including DNN and deep forest better represent biological features for each lncRNA–protein pair.

Table 2 The statistics of LPI data

Dataset	lncRNAs	Proteins	LPIs
Dataset 1	935	59	3479
Dataset 2	885	84	3265
Dataset 3	990	27	4158
Dataset 4	109	35	948
Dataset 5	1704	42	22,133

2 Materials and Methods

2.1 Data Preparation

In this study, five different LPI-related datasets are arranged. Table 2 shows the details of the five datasets. Datasets 1, 2, and 3 contain human LPI data and datasets 4 and 5 contain plant LPI data. Dataset 1 was provided by Li et al. [38]. We obtain 3479 correlations from 935 lncRNAs and 59 proteins after removing lncRNAs and proteins whose sequence information is unknown in the NPInter [39], NONCODE [40], and UniProt [41].

Dataset 2 was built by Zheng et al. [18]. We screen 3265 relationships from 885 lncRNAs and 84 proteins after the preprocessing similar to dataset 1. Dataset 3 was constructed by Zhang et al. [42] and contains 4158 interplays from 990 lncRNAs and 27 proteins.

Datasets 4 and 5 were from Arabidopsis thaliana and Zea mays, respectively. The former contains 948 interactions from 109 lncRNAs and 35 proteins and the latter provides 22,133 associations from 1704 lncRNAs and 42 proteins. Sequence data are extracted from the PlncRNADB database [43] and interaction data are obtained at http://bis.zju.edu.cn/PlncRNADB/.

We represent LPI network as a matrix Y with the element:

$$y_{ij} = \begin{cases} 1, & \text{if lncRNA } l_i \text{ interacts with protein } p_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

2.2 Overview of EnANNDeep

In this study, we develop an ensemble learning framework (EnANNDeep), composed of A*k*NN, DNN, and deep forest, to classify unknown lncRNA–protein pairs. Figure 1 describes the EnANNDeep framework.

As shown in Fig. 1, EnANNDeep mainly contains three procedures after five different LPI datasets are arranged. (1) Feature selection—An ensemble method combining gapped *k*-mer [44], tri-nucleotide composition [45], reverse complement *k*-mer [46], and RNAfold [47] is available for lncRNA feature selection. SSpro [48] and binary profile are used to

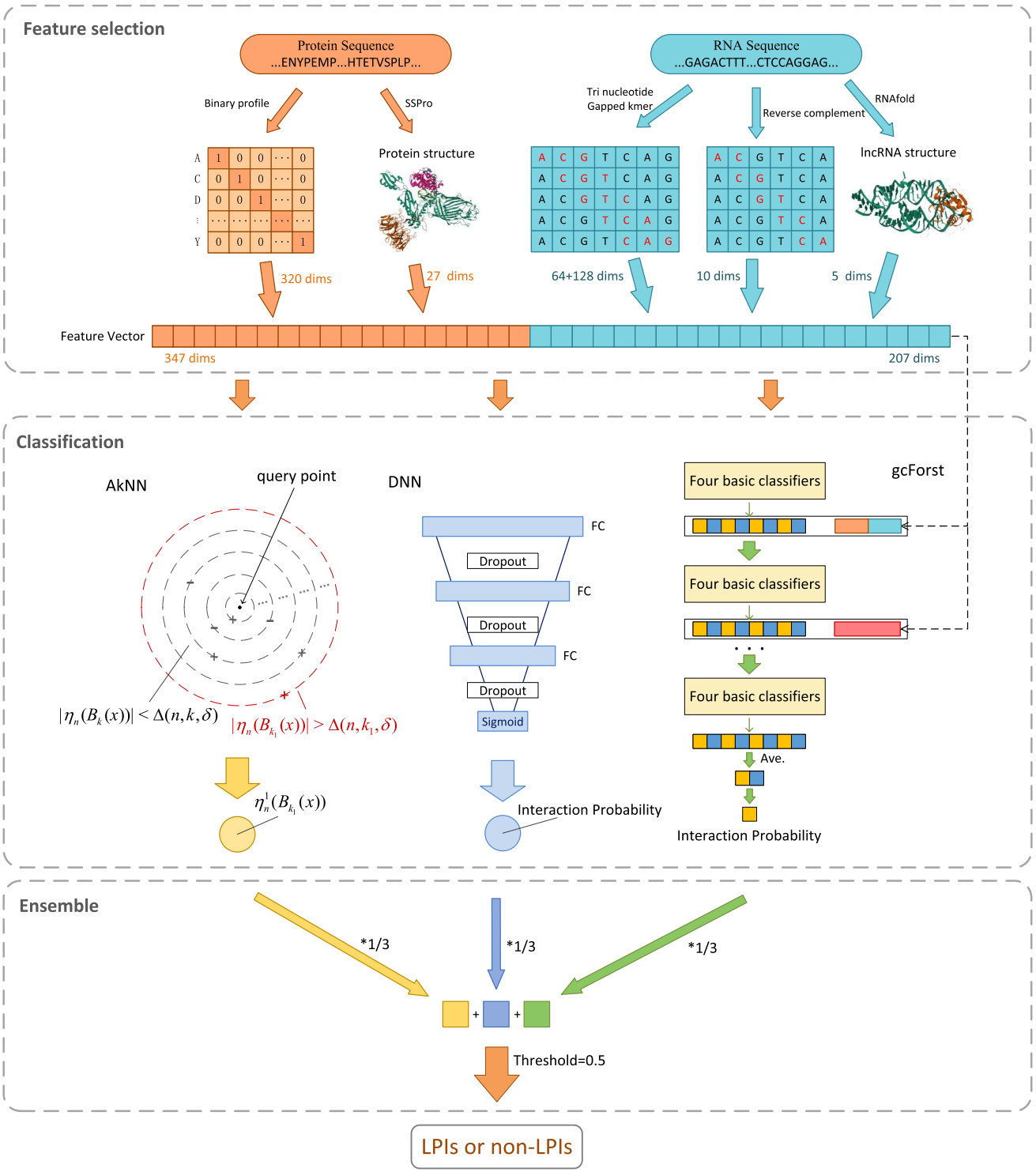


Fig. 1 The flowchart of the LPI-DLDN framework: (1) Feature selection; (2) Classification; (3) Ensemble

chose protein features. (2) Classification—A*k*NN, DNN, and deep forest are exploited to obtain labels of unknown lncRNA–protein pairs, respectively. (3) Ensemble—The results from the above three predictors are integrated based on a soft voting technique.

2.3 Feature Selection

2.3.1 lncRNA Feature Selection

The integration of various lncRNA and protein features contributes to improve LPI prediction accuracy. In this work, an ensemble approach is explored to represent lncRNA features. For given an lncRNA sequence L with length a, where $l_i \in \{A, C, G, T\}$ and $\{i = 1, 2, \ldots, a\}$. EnANNDeep utilizes gapped 3-mer [44], tri-nucleotide composition [45], reverse complement 2-mer [46], and RNAfold [47] to characterize an lncRNA.

The tri-nucleotide composition technique is used to obtain evolutionary features from L. The tri-nucleotide compositions are extracted by scanning the sequence using $\{(1,2,3),(2,3,4),\ldots,(a-2,a-1,a)\}$, where $\{1, 2, 3, \ldots, i, \ldots, a\}$ denotes the i-th nucleotide in L.

The gapped 3-mer method applies 3-mer with gap to obtain local and global information from L. Let b represent the number of non-gapped positions in L, and the number of gaps is $g = 3 - b$. A feature vector of L can be denoted by Eq. (2):

$$f = [u_1, u_2, ..., u_M]^T, \tag{2}$$

where u_i is the number of the i-th gapped 3-mer in L, M is the number of all gapped 3-mers and $M = \binom{3}{b} 4^3$.

The reverse complement 2-mer method is used to extract regulatory features from L. First, 2-mer is generated. Second, reverse complement 2-length contiguous subsequences are eliminated. Finally, the computed occurrence frequencies of the remaining 2-length subsequences are calculated to build an lncRNA feature vector.

The RNA secondary structures have been validated to positively affect protein binding site selection. A dynamic programming technique, RNAfold, is used to infer RNA secondary structures according to its minimum free energy. Five features with high probability structures are extracted by counting occurrence frequency of each unique structure.

2.3.2 Protein Feature Selection

To depict a protein, first, its secondary structures are obtained based on α-helix (H), β-sheet (E), and coil (C) conformation using SSpro [48]. Second, 20 amino acids are divided into three categories based on the computed secondary structures: α-helix contains eight amino acids (E, A, L, M, Q, K, R, and H), β-sheet contains seven amino acids (V, I, Y, C, W, F, and T), and coil contains five amino acids (G, N, P, S, and D). Third, an amino acid can be replaced by its conformation and thus each protein sequence can be represented using H, E or C. 27 3-tuples are obtained from the permutation of the above three conformations. Fourth, 3-tuple is applied to the replaced sequences and the number of each 3-tuple is computed. Finally, the occurrence frequency of each 3-tuple can be calculated by Eq. (3):

$$a_i = \frac{d_i}{a - 3 + 1} (i = 1, 2, \ldots, 27), \tag{3}$$

where d_i is the number of the i-th 3-tuples in L.

In addition, a binary profile describes composition and order of residues in a protein sequence. In this study, a binary profile with a 20×16 dimensions is produced based on a one-hot encoding of 20 amino acids. The details for lncRNA and protein feature extraction are described in Table 3. Thus an lncRNA–protein pair can be represented as a 554-dimensional vector $\boldsymbol{x}$ combining lncRNA and protein features.

2.4 Problem Description

Given an LPI training set $D = (X, Y)$ with labels $\{+1, -1\}$, where a separable metric space $(X, 554)$ denotes the sample space with 554 features and $Y = \{+1, -1\}$ describes the label space. A training example $\boldsymbol{x}$ is a 554-dimensional feature vector applied to characterize an lncRNA–protein pair, $\boldsymbol{y} \in \{+1, -1\}$ denotes its label. The label of $\boldsymbol{x}$ is 1 when there is an interaction between the lncRNA and the protein; the label is -1, otherwise. For any query lncRNA–protein pair $\boldsymbol{x}_i$, we aim to construct an ensemble model, EnANNDeep, to obtain its label.

2.5 Adaptive *k*-Nearest Neighbor

2.5.1 *k*-Nearest Neighbor

k-Nearest Neighbor (*k*-NN) classifier [49] is a simple but effective classification model. It is very appropriate to a classification task where there is lack of prior knowledge about data distribution. The classifier investigates label of a test

Table 3 Numbers of the extracted lncRNA and protein features

	Features	Number
lncRNA	Tri nucleotide	64
	Gapped *k*-mer	128
	Reverse complement *k*-mer	10
	RNAfold	5
Protein	Binary profile	320
	SSpro	27

sample based on the Euclidean distance between the test sample and all training samples.

Given n LPI samples, let $\boldsymbol{x}_i$ $(i = 1, 2, \ldots, n)$ denote the i-th sample with 554 features $(x_{i,1}, x_{i,2}, \ldots, x_{i,554})$. The Euclidean distance between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is represented as:

$$d(x_i, x_j) = \sqrt{(x_{i,1} - x_{j,1})^2 + \cdots + (x_{i,554} - x_{j,554})^2}. \tag{4}$$

Based on the theory provided by Voronoi [50], a Voronoi cell R_i for sample $\boldsymbol{x}_i$ encapsulates its all nearest neighbors and is defined by Eq. (5):

$$R_i = \{\boldsymbol{x}_a \in \mathbb{R}^p \ : \ d(\boldsymbol{x}_a, \boldsymbol{x}_i) \leq d(\boldsymbol{x}_a, \boldsymbol{x}_m), \forall i \neq m\}, \tag{5}$$

where $\boldsymbol{x}_a$ denotes all possible points (samples) within R_i, that is, the nearest neighbors of the example $\boldsymbol{x}_i$.

For any LPI, k-NN classifier determines the nearest samples through the closest edges within the Voronoi cell R_i. A test sample is assigned a label the same as the majority category label of its k nearest training samples based on k-NN classifier.

k-NN uses a fixed radius and can automatically adapt to the variation in marginal distribution. Therefore, it has been broadly applied to various areas. However, the choice of its nearest neighbor number k severely depends on features of each neighborhood and thus may greatly vary between different points. In the input space, for the regions where conditional expectation of $\boldsymbol{x}$ tends to 0, larger k is required for accurate prediction. For other regions where the conditional expectation is $+1$ or -1, smaller k can satisfy the requirement and larger k may result in incorrect classification due to the inconsistence of labels in the neighboring regions. Thus k-NN classifier has to select a single value for k to trade off the above two situations. To solve this problem, AkNN classifier is designed to separately select the right k for each neighborhood.

2.5.2 Adaptive *k*-Nearest Neighbor

Inspired by the AkNN algorithm proposed by Balsubramani et al. [51], we design an AkNN algorithm to compute interaction probability for each lncRNA–protein pairs. For a training set $(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n) \in \mathcal{X} \times \mathcal{Y}$, let all LPI data draw from an unobserved independent identically distribution P on $\mathcal{X} \times \mathcal{Y}$. Let μ represent the marginal distribution on $\mathcal{X}$: if (X, Y) denotes a random draw from P, let $\eta(\boldsymbol{x}) = \mathrm{E}(Y|X = \boldsymbol{x})$, then for any measurable set $S \subseteq \mathcal{X}$:

$$\mu(S) = \Pr(X \in S). \tag{6}$$

For any given sample $\boldsymbol{x} \in \mathcal{X}$, conditional expectation of Y can be denoted by Eq. (7):

$$\eta(\boldsymbol{x}) = E(Y|X = \boldsymbol{x}) \in [-1, 1]. \tag{7}$$

For any S where $\mu(S) > 0$ and given $X \in S$, conditional expectation of Y can be described by Eq. (8):

$$\eta(S) = E(Y|X \in S) = \frac{1}{\mu(S)} \int_S \eta(\boldsymbol{x}) \mathrm{d}\mu(\boldsymbol{x}). \tag{8}$$

Thus the error risk of k-NN classifier: $g : \mathcal{X} \to \{-1, +1\}$ is the probability that it incorrectly classifies a query sample on the training set $(X, Y) \sim P$. The risk is denoted by Eq. (9):

$$R(g) = P(\{(\boldsymbol{x}, \boldsymbol{y}) \ : \ g(\boldsymbol{x}) \neq \boldsymbol{y}\}). \tag{9}$$

For $\boldsymbol{x} \in \mathcal{X}$ and $r > 0$, let $B(\boldsymbol{x}, r)$ represent the closed ball with radius r centered at $\boldsymbol{x}$:

$$B(\boldsymbol{x}, r) = \{z \in \mathcal{X} : d(\boldsymbol{x}, z) \leq r\}. \tag{10}$$

For a query lncRNA–protein pair $\boldsymbol{x}$, AkNN classifier predicts its label based on the training lncRNA–protein pairs closest to $\boldsymbol{x}$. The empirical count is defined by Eq. (11):

$$\#_n(S) = |\{i : \boldsymbol{x}_i \in S\}|. \tag{11}$$

The probability mass can be described by Eq. (12):

$$\mu_n(S) = \frac{\#_n(S)}{n}. \tag{12}$$

When the empirical count is non-zero, the empirical bias can be defined by Eq. (13).

$$\eta_n(S) = \eta_n^y(S) - \frac{1}{|Y|}, \tag{13}$$

where n indicates the number of all lncRNA–protein pairs. $|Y|$ denotes the number of classes. In this manuscript, $|Y|$ is 2.

$$\eta_n^y(S) = \frac{\#_n\{\boldsymbol{x}_i \in S \text{ and } \boldsymbol{y}_i = y\}}{\#_n(S)}. \tag{14}$$

AkNN classification model is described in Algorithm 1. The label of a query lncRNA–protein pair $\boldsymbol{x}$ can be predicted through expanding a ball around $\boldsymbol{x}$ until it produces a significant bias based on Algorithm 1.

Algorithm 1: The adaptive k-nearest neighbor classifier

Input: A training set $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n) \in \mathcal{X} \times \{+1, -1\}$, a confident parameter $0 \leq \delta \leq 1$

Output: The label of a query lncRNA-protein pair $\boldsymbol{x}$

Step 1 For any integer k, assume that $B_k(\boldsymbol{x})$ represent the smallest ball which is centered at x and precisely contains k training samples.

Step 2 Find the smallest $0 < k \leq n$ such that $B_k(\boldsymbol{x})$ has a significant bias: $|\eta_n(B_k(\boldsymbol{x}))| > \Delta(n, k, \delta)$ where $\Delta(n, k, \delta) = c_1 \sqrt{\frac{\log n + \log(1/\delta)}{k}}$

Step 3 Return the label $\arg\max_y \eta_n(B_k(\boldsymbol{x}))$ when such a ball exists or $\eta_n(B_k(\boldsymbol{x}))$ is largest.

In Algorithm 1, $\Delta(n,k,\delta)$ denotes a confidence interval of average labels in the region closest to the query sample $\boldsymbol{x}$. c_1 represents a constant.

Algorithm 1 infinitely makes many parameter selection. It picks k for each query point and asks for a single failure probability to measure how to assign its confidence intervals. In comparing to standard k-NN classifier, the AkNN classification algorithm seems to merely replace the parameter k with another parameter δ. However, it is not accurate. δ, a customary confident level parameter, provides an upper bound upon the failure probability for Algorithm 1.

To simplify the parameters, we replace Δ with $\Delta = \frac{A}{\sqrt{k}}$ in $\Delta(n,k,\delta) = c_1\sqrt{\frac{\log n+\log(1/\delta)}{k}}$. The parameter A is used to control conservations in Algorithm 1 and $A \to 0$ denotes the most aggressive setting where Algorithm 1 never abstains. The detailed discussion is provided by Balsubramani et al. [51].

2.5.3 Deep Neural Network

The rapid development of machine learning models and computer hardware promotes the birth of DNNs. DNN is a feed-forward artificial neural networks. A DNN consists of one input layer, multiple hidden layers composed of nonlinear hidden units, and one larger output layer. The input layer achieves the original data. Each hidden unit j in a hidden layer uses an activation function to map the input x_j from the input layer to a scalar state. The output layer accommodates multiple hidden Markov model states.

DNNs have been already broadly applied to various association prediction [28]. For example, Zhao et al. [52] identified drug–target interactions combining graph convolutional network and DNN. Chu et al. [29] developed an optimized DNN to screen epidermal growth factor receptor inhibitors. Wang et al. [53] exploited a deep convolutional neural network-based drug–target interactions algorithm. Wei et al. [35] designed a DNN-based lncRNA-disease association prediction approach.

In this study, we utilize DNN to reveal possible LPIs. The DNN-based LPI prediction framework is shown in Fig. 2.

In the DNN model, the input layer has 554 neurons and achieves the input LPI samples with 554-dimensional features. The following two layers are hidden layers. The two layers are full connection layers containing 128 and 64 neurons, respectively. And each hidden layer follows by a dropout layer with the rate of 0.5 to avoid over-fitting by setting the output of 50% units to 0. Exponential Linear Unit (ELU) is considered as an activation function in the hidden layers. ELU can alleviate gradient vanishing, make the average output of an activation unit closer to 0 to achieve the effect of batch normalization and reduce the computation time. In addition, ELU is only qualitative but not quantitative for the input characteristics because it is an exponential function when it is negative. More importantly, ELU contributes to faster learning and better generalization ability on DNNs. It is denoted by Eq. (15):

$$y_i = \begin{cases} a(e^{x_i}-1) & \text{if} \quad (x_i < 0) \\ x_i & \text{if} \quad (x_i \geq 0) \end{cases} \tag{15}$$

Our objective is to quantify how many the predicted labels differ from the real ones by minimizing the binary cross-entropy in the process of training by Eq. (16):

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]. \tag{16}$$

where $\boldsymbol{y}_i$ is the true label and $\hat{\boldsymbol{y}}_i$ denotes the probability that the i-th sample is predicted to be positive LPI. The training is implemented with 100 epochs and each epoch has a mini-batch with the size of 128 to update its weights. We use the Adam algorithm [54] as the optimization technique to train DNN.

The final output layer contains a single neuron to output an interaction probability for each query lncRNA–protein pair based on a sigmoid function defined by Eq. (17):

$$y_i = \frac{1}{1+e^{-x_i}}. \tag{17}$$

The sigmoid function can map a real number to the interval of (0,1). It is smooth and easy to derivation and is thus used as an activation function in the output layer of DNN to compute interaction score for each lncRNA–protein pair.

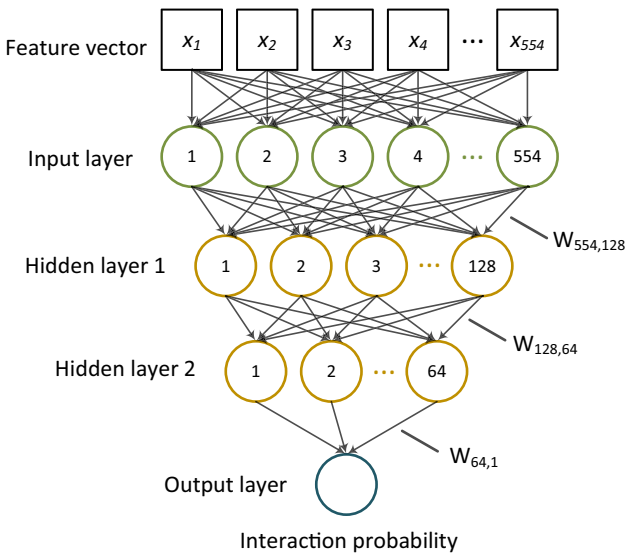


Fig. 2 The flowchart of DNN-based LPI prediction algorithm

2.5.4 Deep Forest

To tackle complicated tasks, learning models gradually go deep [55]. However, traditional deep algorithms are always designed based on neural networks. Non-neural network style-based deep models will demonstrate great learning ability if they can go deep, especially when neural networks are multi-layered deep models with parameterized differentiable nonlinear modules. Considering this feature of neural networks, deep forest [56, 57], a non-neural network style deep model, is built upon multi-grained cascade framework.

Deep forest is a novel ensemble algorithm. Its feature learning capability is further boosted by multi-grained scanning the input data. Second, its complexity can be automatically set. Third, it performs better even on small-scale data. Finally, the training costs can be controlled based on available computational resources. Deep forest only needs to train much fewer hyper-parameters in comparing to other deep learning models. Therefore, deep forest obtains highly competitive classification ability while its training time drops sharply.

In this manuscript, deep forest with no more than 20 layers is utilized to classify unobserved lncRNA–protein pairs. Random forest [58, 59] and Extra trees [60] are chosen as basic classifiers. The random forest technique [58, 59] is a general-purpose, nonparametric, and interpretable classification model. It is an ensemble of a few randomized decision trees and can return measurements of variable importance. It has unique characteristics in dealing with complex data structures, small sample size, and high-dimensional feature space. In particular, it demonstrates excellent performance when the number of variables is far more than the number of samples.

The Extra tree model [60] is an ensemble of unpruned decision trees based on the classical top–down procedure. Extra tree has three advantages: First, it splits nodes by fully randomly selecting cut-points and contributes to more strongly reduce variance than the weaker randomization algorithms. Second, it utilizes the whole learning samples rather than a bootstrap replicas to minimize classification bias. Finally, it contains a node splitting scheme to obtain much smaller constant factor during cut-point optimization.

In the proposed deep forest model, each cascade layer consists of two random forests and two Extra trees. Each estimator consists of 100 decision trees. In each layer, for a given LPI feature, each classifier calculates the ratio of the feature belonging to positive class or negative class. The predicted class probability from all classifiers forms a class vector. The vector is concatenated with the raw LPI feature vector as input in the next level.

As illustrated in Fig. 3, a 554-dimensional vector is taken as input of deep forest. After training four basic classifiers, an 8-dimensional class vector is produced and concatenated with the 554-dimensional vector to generate a 562-dimensional feature vector. The produced vector is considered as input in the second layer. Similar to the first layer, the second layer of deep forest also generates another 562-dimensional vector applied to the third layer. If the estimated performance outperforms all previously-constructed layers, deep forest continues to increase a new layer. The model will

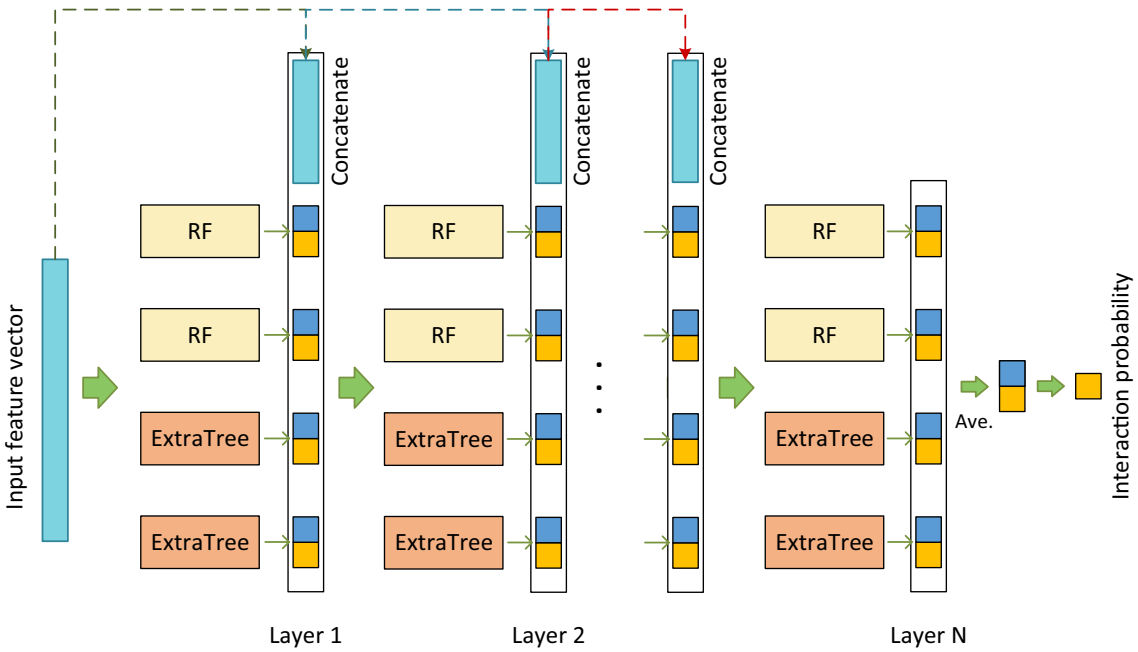


Fig. 3 The flowchart of deep forest

terminate training when its performance fails to improve in the successive two layers. Finally, in the output layer, for each lncRNA–protein pair, its predicted interaction probability belonging to positive class and negative class is averaged, respectively. The class that the lncRNA–protein pair has higher average interaction probability is chosen as the final class.

In particular, similar to DNN, deep forest utilizes a cascade structure. In the structure, each level receives features from its preceding level, and outputs the results to next level. Therefore, although the proportion of an 8-dimensional class vector in the input layer may be relatively smaller, its proportion in an LPI feature vector will continuing increase with the deepening of the number of layers. Therefore, in our model, the 8-dimensional class vector cannot be drown out.

2.5.5 Ensemble Learning

Ensemble learning demonstrates better prediction accuracy of a single model through training multiple classifiers and integrating their predictions [27, 61, 62]. Chen et al. [63] exploited a decision tree ensemble algorithm to uncover possible miRNA-disease associations. Zhang et al. [23] designed a sequence feature projection-based ensemble learning model to identify LPI candidates. Yi et al. [27] exploited a stacking ensemble learning algorithm to discover ncRNA–protein interactions.

Although A*k*NN, DNN, and deep forest can effectively predict LPIs, their predictive performance remains improvement. In this study, we present a soft voting-based ensemble learning framework, composed of A*k*NN, DNN, and deep forest, to enhance the classification ability of existing single model. Let S_{AkNN}, S_{DNN}, and S_{DF} denote association probability of an lncRNA–protein pair obtained by A*k*NN, DNN, and deep forest, respectively, its final relevance score is defined by Eq. (18) based on a soft voting technique:

$$S = \frac{1}{3}S_{AkNN} + \frac{1}{3}S_{DNN} + \frac{1}{3}S_{DF.} \tag{18}$$

An lncRNA–protein pair is labeled as positive class if its score is larger than 0.5 based on Eq. (18); otherwise, the lncRNA–protein pair is classified to negative.

3 Results

3.1 Evaluation Metrics

In the experiments, precision, recall, accuracy, F1 score, AUC and AUPR are applied to assess the performance of EnANNDeep. For the six measurements, higher values indicate better prediction ability. The experiments are repeatedly implemented for 20 times and the average values on the 20 rounds are selected as the final performance.

3.2 Experimental Settings

We conduct grid search to find the optimal parameters in SFPEL, PMDKN, CatBoost, PLIPCOM, and EnANNDeep when the five LPI prediction approaches obtain the best performance. The details are listed in Table 4. The parameters in LPI-SKF are set to default values provided by Zhou et al. [17].

In addition, to investigate the prediction performance of EnANNDeep for a new lncRNA or protein, three different fivefold CVs are designed.

1. Fivefold CV on lncRNAs (CV_l): rows in Y are randomly hidden for testing, that is, 80% of lncRNAs are randomly chosen as a training set and the remaining 20% is used as a testing set in each round. CV_l is used to find interacting proteins for a new lncRNA without any associated proteins.
2. Fivefold CV on proteins (CV_p): columns in Y are randomly hidden for testing, that is, 80% of proteins are randomly chosen as a training set and the remaining 20% is used as a testing set in each round. CV_p is used to identify interacting lncRNAs for a new protein without any associated lncRNAs.
3. Fivefold CV on lncRNA–protein pairs (CV_{lp}): lncRNA–protein pairs in Y are randomly hidden for testing, that is, 80% of lncRNA–protein pairs are chosen as a training

Table 4 Parameter settings

Method	Parameter settings
SFPEL	$\mu = 0.001$, $\lambda = 0.0001$, $\gamma = 4$
PMDKN	$\eta = 5$, $\lambda = 1$, $\gamma = 1$, $\mu = 100$
	$\Upsilon = 100$, $\delta = 2$,
CatBoost	Iterations = 2, learning_rate = 0.5,
	logging_level = None, depth = 5
PLIPCOM	learning_rate = 1, n_estimators = 100,
	max_depth = 3, min_samples_leaf = 10,
	min_samples_split = 2, max_features = 30,
	random_state = 10
EnANNDeep	A*k*NN: log_complexity = 1.0;
	DNN: number of layer = 4, dropout = 0.5,
	Activation ='elu', epochs = 100,
	Optimizer = Adam(lr = 1e−4);
	Learning rate = 0.001
	Deep forest: loss ='binary_crossentropy'
	max_layers = 20, n_trees = 100
	max_depth = None, batch_size = 128,
	n_estimators = 2

set and the remaining 20% is used as a testing set in each round. CV_{lp} is used to uncover interaction information based on known LPIs.

3.3 Comparison with Five State-of-the-Art LPI Prediction Methods

We compare the proposed EnANNDeep method with five representative LPI prediction methods (SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF) to measure the prediction performance of EnANNDeep. SFPEL is an ensemble learning method for LPI prediction based on sequence feature projection. SFPEL first extracted sequence features for lncRNAs and proteins and then computed lncRNA similarity and protein similarity. Finally, it used a feature projection-based ensemble learning framework to predict LPIs combining the computed similarity matrices.

PMKDN is a neighborhood nonnegative matrix decomposition model applied to possible LPI inference. PMKDN first selected multiple biological features of lncRNAs and proteins. Second, it combined protein GO ontology annotation and sequences, lncRNA sequences, and modified LPI network to calculate lncRNA similarity and protein similarity. Finally, it utilized a projection-based neighborhood nonnegative matrix decomposition algorithm to infer potential LPIs.

CatBoost is a new gradient boosting algorithm. CatBoost implemented two key techniques, that is, ordered boosting which is a permutation-driven alternative to a classification model, and a categorical feature procession strategy. The combination of them promotes CatBoost to outperform the other available boosting techniques. CatBoost has been applied to LPI discovery and obtained better LPI classification ability.

PLIPCOM employed two network features, diffusion features and HeteSim features, and built an LPI prediction model integrating the Gradient Tree Boosting (GTB) algorithm.

LPI-SKF first computed lncRNA similarity based on expression profiles and sequences of lncRNAs and LPI network, and protein similarity based on statistical features and sequences of proteins and LPI network. It then constructed a universal similarity kernel matrix for new LPI identification based on a similarity kernel fusion technique.

We evaluate the performance of our proposed EnANNDeep framework under three different fivefold CVs. During CVs, we randomly select unknown lncRNA–protein pairs as negative samples (non-LPIs). To reduce the overfitting problem produced by data imbalance, we set the ratio of negative LPIs to known LPIs as 1. That is, the number of the screened negative LPIs is the same as one of observed LPIs in the divided training set and test set. The best measurements are represented as bold in each row in Tables 5, 6 and 7.

Table 5 illustrates the prediction results from six LPI identification models in terms of the above six evaluation metrics under CV_l. EnANNDeep achieves the highest average precision, recall, accuracy, F1 score, AUC, and AUPR. In particular, compared to SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF, the average AUC computed by EnANNDeep outperforms 32.92%, 17.29%, 12.76%, 7.99%, and 3.94%, respectively. The average AUPR calculated by EnANNDeep are better 33.33%, 15.85%, 12.78%, 9.76%, and 5.29% than the above five methods. The result suggest that EnANNDeep may be suitable to linkage discovery for a new lncRNA.

Table 6 describes the six evaluation values under CV_p. From Table 6, it can be found that EnANNDeep computes the best average precision, recall, accuracy, AUC and AUPR under CV_p. Although EnANNDeep calculates relatively lower F1 score, it greatly boosts the precision, recall, accuracy, AUC, and AUPR performance. For example, compared to SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF, its AUC boosts 42.76%, 22.23%, 31.74%, 21.79%, and 25.74%, respectively, AUPR improves 36.15%, 14.68%, 31.87%, 23.25%, and 18.82%, respectively. AUC and AUPR can more representatively characterize the performance of classifiers compared to the other four measurements. EnANNDeep distinctly outperforms the other five algorithms in terms of AUC and AUPR. Therefore, it is appropriate to prioritize potential lncRNAs for a new protein.

The experimental results under CV_{lp} are listed in Table 7. The results illustrate the optimal LPI classification ability of EnANNDeep. Under CV_{lp}, EnANNDeep obtains the best average recall, accuracy, F1 score, AUC, and AUPR. For example, it computes F1 score of 0.8569, which is 9.46%, 30.93%, 8.51%, 3.09%, and 18.09% better than SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF, respectively. The computed average AUC outperforms 7.07%, 17.53%, 6.55%, 2.43%, and 1.13%, respectively, and AUPR is better 4.67%, 14.98%, 4.19%, 3.74%, and 4.83%, respectively. SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF are state-of-the-art LPI prediction algorithms. EnANNDeep greatly outperforms the five methods. The comparative results suggest the powerful performance of EnANNDeep under CV_{lp}. That is, EnANNDeep can more accurately mine underlying relationships between lncRNAs and proteins even in the absence of some LPIs.

3.4 Comparison of Different Voting Methods

We conduct several experiments to observe the affect of voting techniques on the classification performance. We consider two voting techniques: soft voting approach and hard voting approach. Given an unobserved lncRNA–protein pair,

Table 5 The performance of five LPI prediction methods on CV_l

Metric	Dataset	SFPEL	PMDKN	CatBoost	PLIPCOM	LPI-SKF	EnANNDeep
Precision	Dataset 1	0.5984 ± 0.0088	0.7583 ± 0.0100	0.8639 ± 0.0047	0.8335 ± 0.0185	**0.8757 ± 0.0086**	0.8364 ± 0.0042
	Dataset 2	0.4822 ± 0.0139	0.7319 ± 0.0159	0.8607 ± 0.0054	0.8584 ± 0.0124	**0.8627 ± 0.0223**	0.8550 ± 0.0038
	Dataset 3	0.5782 ± 0.0111	0.6581 ± 0.0167	**0.7505 ±0.0114**	0.6846 ± 0.0169	0.7298 ± 0.0153	0.6979 ± 0.0056
	Dataset 4	0.5114 ± 0.0242	0.5396 ± 0.0108	0.5171 ± 0.0387	0.5390 ± 0.0924	0.6108 ± 0.0249	**0.6882 ± 0.0126**
	Dataset 5	0.5908 ± 0.0028	0.6454 ± 0.0127	0.5571 ± 0.0058	0.7732 ± 0.0197	0.7517 ± 0.0098	**0.8035 ± 0.0014**
	Ave.	0.5522	0.6667	0.7099	0.7377	0.7661	**0.7762**
Recall	Dataset 1	0.5732 ± 0.0050	0.6763 ± 0.0074	0.8692 ± 0.0138	0.8397 ± 0.0220	0.5932 ± 0.0156	**0.9700 ± 0.0030**
	Dataset 2	0.5125 ± 0.0079	0.6484 ± 0.0165	0.8678 ± 0.0153	0.8789 ± 0.0173	0.5212 ± 0.0107	**0.9707 ± 0.0028**
	Dataset 3	0.5534 ± 0.0078	0.6017 ± 0.0105	0.6608 ± 0.0239	0.6680 ± 0.0226	0.6226 ± 0.0058	**0.8028 ± 0.0041**
	Dataset 4	0.4949 ± 0.0171	0.5195 ± 0.0060	0.4173 ± 0.0475	0.3744 ± 0.0884	**0.6056 ± 0.0280**	0.5628 ± 0.0297
	Dataset 5	0.5672 ± 0.0011	0.5975 ± 0.0100	0.5870 ± 0.0313	0.7803 ± 0.0320	0.6727 ± 0.0037	**0.8616 ± 0.0024**
	Ave.	0.5402	0.6087	0.6804	0.7083	0.6030	**0.8336**
Accuracy	Dataset 1	0.5752 ± 0.0059	0.6759 ± 0.0074	0.8659 ± 0.0053	0.8359 ± 0.0133	0.7254 ± 0.0032	**0.8900 ± 0.0033**
	Dataset 2	0.5122 ± 0.0074	0.6480 ± 0.0165	0.8634 ± 0.0060	0.8670 ± 0.0099	0.7065 ± 0.0081	**0.9029 ± 0.0031**
	Dataset 3	0.5547 ± 0.0083	0.6014 ± 0.0105	0.7190 ± 0.0043	0.6801 ± 0.0134	0.6544 ± 0.0092	**0.7275 ± 0.0045**
	Dataset 4	0.4963 ± 0.0171	0.5181 ± 0.0060	0.5210 ± 0.0286	0.5279 ± 0.0526	0.5727 ± 0.0196	**0.6530 ± 0.0122**
	Dataset 5	0.5639 ± 0.0021	0.5974 ± 0.0100	0.5604 ± 0.0061	0.7758 ± 0.0158	0.6726 ± 0.0036	**0.8253 ± 0.0007**
	Ave.	0.5405	0.6082	0.7059	0.7373	0.6663	**0.7997**
F1 score	Dataset 1	0.5272 ± 0.0056	0.6393 ± 0.0071	0.8660 ± 0.0064	0.8364 ± 0.0140	0.6298 ± 0.0070	**0.8982 ± 0.0029**
	Dataset 2	0.4563 ± 0.0081	0.6135 ± 0.0150	0.8636 ± 0.0071	0.8684 ± 0.0104	0.5828 ± 0.0117	**0.9091 ± 0.0029**
	Dataset 3	0.5072 ± 0.0083	0.5609 ± 0.0109	0.7007 ± 0.0097	0.6759 ± 0.0147	0.5950 ± 0.0086	**0.7465 ± 0.0033**
	Dataset 4	0.4493 ± 0.0175	0.4739 ± 0.0064	0.4565 ± 0.0427	0.4340 ± 0.0758	0.5401 ± 0.0232	**0.6149 ± 0.0202**
	Dataset 5	0.5181 ± 0.0018	0.5550 ± 0.0102	0.5690 ± 0.0151	0.7763 ± 0.0193	0.6345 ± 0.0041	**0.8314 ± 0.0008**
	Ave.	0.4916	0.5685	0.6912	0.7182	0.5964	**0.8000**
AUC	Dataset 1	0.6503 ± 0.0117	0.8518 ± 0.0148	0.9336 ± 0.0029	0.8972 ± 0.0222	**0.9344 ± 0.0073**	0.9330 ± 0.0023
	Dataset 2	0.5243 ± 0.0148	0.7959 ± 0.0330	0.9250 ± 0.0036	0.9196 ± 0.0108	0.9199 ± 0.0149	**0.9487 ± 0.0021**
	Dataset 3	0.6093 ± 0.0166	0.7028 ± 0.0210	0.8050 ± 0.0037	0.7571 ± 0.0137	0.8117 ± 0.0159	**0.8199 ± 0.0030**
	Dataset 4	0.4927 ± 0.0342	0.5362 ± 0.0120	0.5212 ± 0.0340	0.5549 ± 0.0622	0.6479 ± 0.0379	**0.7189 ± 0.0104**
	Dataset 5	0.6279 ± 0.0041	0.6949 ± 0.0201	0.5926 ± 0.0087	0.8551 ± 0.0149	0.8455 ± 0.0076	**0.9093 ± 0.0008**
	Ave.	0.5809	0.7163	0.7555	0.7968	0.8319	**0.8660**
AUPR	Dataset 1	0.6203 ± 0.0125	0.8521 ± 0.0147	**0.9209 ±0.0038**	0.8544 ± 0.0340	0.9196 ± 0.0092	0.9154 ± 0.0043
	Dataset 2	0.4976 ± 0.0126	0.8150 ± 0.0223	0.9144 ± 0.0054	0.8832 ± 0.0189	0.8787 ± 0.0260	**0.9350 ± 0.0030**
	Dataset 3	0.5897 ± 0.0118	0.6989 ± 0.0224	0.7907 ± 0.0053	0.7201 ± 0.0193	0.7772 ± 0.0198	**0.8014 ± 0.0040**
	Dataset 4	0.5375 ± 0.0263	0.5487 ± 0.0122	0.5346 ± 0.0315	0.5641 ± 0.0779	0.6348 ± 0.0340	**0.7133 ± 0.0092**
	Dataset 5	0.6033 ± 0.0033	0.6809 ± 0.0169	0.5658 ± 0.0074	0.8335 ± 0.0204	0.8364 ± 0.0170	**0.9073 ± 0.0012**
	Ave.	0.5697	0.7191	0.7453	0.7711	0.8093	**0.8545**

the hard voting method first obtains label of an lncRNA–protein pair based on classification results from A*k*NN, DNN, and deep forest, respectively. Hard voting then classifies the sample as positive if its classification results from no less than two basic predictors are positive; otherwise, the pair is labeled as a negative class. The comparison results of two voting approaches under three CVs are shown in Tables 8, 9 and 10. From Tables 8, 9 and 10, we can find that the soft voting-based ensemble learning model can obtain better performance compared to the hard voting method.

3.5 The Effect of Numbers of RNA Secondary Structures on the Performance

Although abundant biological information contributes to improve LPI prediction performance, the biological features exist information robust and increase computational complexity. Therefore, we select the representative features to describe lncRNA secondary structures. Tables 11, 12 and 13 list the performance of EnANNDeep based on 5, 10, 64, and 128 RNA secondary structures with high probability. The results indicate that five features with high probability

Table 6 The performance of six LPI prediction methods on CV_p

Metric	Dataset	SFPEL	PMDKN	CatBoost	PLIPCOM	LPI-SKF	EnANNDeep
Precision	Dataset 1	0.5548 ± 0.0175	0.6814 ± 0.0407	0.2462 ± 0.1042	0.5950 ± 0.2572	0.7009 ± 0.1208	**0.8758 ± 0.0547**
	Dataset 2	0.5277 ± 0.0263	0.6834 ± 0.0457	0.0826 ± 0.0924	0.5217 ± 0.2076	0.6138 ± 0.1316	**0.8885 ± 0.0372**
	Dataset 3	0.5460 ± 0.0437	0.6428 ± 0.0361	0.4337 ± 0.1142	0.5476 ± 0.1361	0.6639 ± 0.1119	**0.7420 ± 0.0313**
	Dataset 4	0.5350 ± 0.0406	0.6120 ± 0.0090	0.6636 ± 0.0158	0.7458 ± 0.0591	0.7261 ± 0.0412	**0.7667 ± 0.0133**
	Dataset 5	0.4996 ± 0.0001	0.6957 ± 0.0180	0.7300 ± 0.0096	0.8153 ± 0.0476	0.7264 ± 0.1465	**0.8381 ± 0.0053**
	Ave.	0.5326	0.6631	0.4312	0.6451	0.6862	**0.8222**
Recall	Dataset 1	0.5164 ± 0.0285	**0.6043 ±0.0300**	0.1920 ± 0.1085	0.1925 ± 0.1052	0.5415 ± 0.0702	0.4451 ± 0.1061
	Dataset 2	0.4895 ± 0.0343	**0.5960 ±0.0366**	0.0306 ± 0.0477	0.1063 ± 0.0883	0.4114 ± 0.0551	0.3573 ± 0.0564
	Dataset 3	0.5036 ± 0.0285	0.5742 ± 0.0290	0.3430 ± 0.1116	0.4181 ± 0.1523	0.4982 ± 0.0746	**0.6774 ± 0.0426**
	Dataset 4	0.5227 ± 0.0121	0.5572 ± 0.0077	0.6878 ± 0.0204	0.7151 ± 0.0833	0.5402 ± 0.0415	**0.7495 ± 0.0233**
	Dataset 5	0.4586 ± 0.0106	0.6274 ± 0.0156	0.7981 ± 0.0191	0.7915 ± 0.0661	0.5811 ± 0.0589	**0.8521 ± 0.0063**
	Ave.	0.4982	0.5918	0.4103	0.4447	0.5145	**0.6163**
Accuracy	Dataset 1	0.5594 ± 0.0167	0.6037 ± 0.0300	0.4334 ± 0.0534	0.5867 ± 0.0757	0.5794 ± 0.1383	**0.6968 ± 0.0505**
	Dataset 2	0.5317 ± 0.0278	0.5952 ± 0.0368	0.4990 ± 0.0234	0.5527 ± 0.1638	0.5220 ± 0.0482	**0.6580 ± 0.0279**
	Dataset 3	0.5499 ± 0.0453	0.5739 ± 0.0290	0.5018 ± 0.0640	0.5443 ± 0.0760	0.5584 ± 0.0777	**0.7147 ± 0.0267**
	Dataset 4	0.5285 ± 0.0276	0.5558 ± 0.0077	0.6685 ± 0.0146	0.7318 ± 0.0391	0.6202 ± 0.0332	**0.7595 ± 0.0121**
	Dataset 5	0.4992 ± 0.0003	0.6274 ± 0.0156	0.7510 ± 0.0128	0.8041 ± 0.0253	0.6636 ± 0.0644	**0.8433 ± 0.0029**
	Ave.	0.5337	0.5912	0.5707	0.6425	0.5902	**0.7345**
F1 score	Dataset 1	**0.6877 ± 0.0091**	0.5686 ± 0.0312	0.2086 ± 0.1032	0.2697 ± 0.1469	0.5399 ± 0.0745	0.5436 ± 0.1086
	Dataset 2	**0.6770 ± 0.0134**	0.5637 ± 0.0376	0.0402 ± 0.0607	0.1592 ± 0.1078	0.4092 ± 0.0634	0.4656 ± 0.0661
	Dataset 3	0.6908 ± 0.0281	0.5377 ± 0.0295	0.3703 ± 0.1110	0.4435 ± 0.1063	0.4929 ± 0.0804	**0.6965 ± 0.0274**
	Dataset 4	0.6736 ± 0.0113	0.5185 ± 0.0067	0.6737 ± 0.0143	0.7248 ± 0.0396	0.5468 ± 0.0408	**0.7554 ± 0.0153**
	Dataset 5	0.6659 ± 0.0001	0.5878 ± 0.0152	0.7617 ± 0.0138	0.8002 ± 0.0299	0.5908 ± 0.0734	**0.8446 ± 0.0030**
	Ave.	**0.6790**	0.5553	0.4109	0.4795	0.5159	0.6611
AUC	Dataset 1	0.5361 ± 0.0538	0.7074 ± 0.0601	0.4365 ± 0.0682	0.6163 ± 0.1158	0.6293 ± 0.1142	**0.9093 ± 0.0106**
	Dataset 2	0.4815 ± 0.0688	0.6903 ± 0.0736	0.5076 ± 0.0690	0.5525 ± 0.1253	0.5235 ± 0.0899	**0.9175 ± 0.0134**
	Dataset 3	0.5279 ± 0.0418	0.6477 ± 0.0581	0.4971 ± 0.0740	0.5633 ± 0.0750	0.5848 ± 0.1577	**0.7907 ± 0.0247**
	Dataset 4	0.5489 ± 0.0131	0.6116 ± 0.0154	0.7250 ± 0.0148	0.8067 ± 0.0469	0.7202 ± 0.0571	**0.8422 ± 0.0090**
	Dataset 5	0.4173 ± 0.0213	0.7548 ± 0.0313	0.8286 ± 0.0104	0.8929 ± 0.0225	0.8000 ± 0.1136	**0.9280 ± 0.0018**
	Ave.	0.5023	0.6824	0.5990	0.6863	0.6516	**0.8775**
AUPR	Dataset 1	0.5642 ± 0.0491	0.7727 ± 0.0523	0.3983 ± 0.0474	0.5637 ± 0.2076	0.7347 ± 0.1155	**0.8775 ± 0.0150**
	Dataset 2	0.5330 ± 0.0588	0.7766 ± 0.0568	0.4595 ± 0.0787	0.5028 ± 0.1684	0.5965 ± 0.1215	**0.8803 ± 0.0193**
	Dataset 3	0.5522 ± 0.0523	0.7064 ± 0.0469	0.5484 ± 0.0683	0.5611 ± 0.1209	0.6556 ± 0.1277	**0.7732 ± 0.0251**
	Dataset 4	0.6338 ± 0.0174	0.6458 ± 0.0111	0.7201 ± 0.0183	0.8064 ± 0.0339	0.7415 ± 0.0543	**0.8444 ± 0.0118**
	Dataset 5	0.4610 ± 0.0201	0.7650 ± 0.0222	0.8016 ± 0.0140	0.8644 ± 0.0368	0.7600 ± 0.1657	**0.9221 ± 0.0022**
	Ave.	0.5488	0.7333	0.5856	0.6597	0.6977	**0.8595**

structures can accurately depict RNA secondary structures. Therefore, we chose five lncRNA secondary structures to reduce computation cost.

3.6 Case Study

In this section, we implement several cases to further evaluate the performance of EnANNDeep. We run the experiments for ten times and compute the average performance from the ten time results.

3.6.1 Finding Interacting Proteins for New lncRNAs

LncRNA Small Nucleolar RNA Host Gene 1 (SNHG1) has close linkage with multiple human diseases. For example, SNHG1 is up-regulated in gastric cancer and may serve as a potential therapeutic target of gastric cancer [64]. It promotes cell proliferation and cell cycle progression carcinoma and inhibits cell apoptosis in hepatocellular carcinoma [65]. It also enhances neuroinflammation in Parkinson's

Table 7 The performance of six LPI prediction methods on CV_{lp}

Metric	Dataset	SFPEL	PMDKN	CatBoost	PLIPCOM	LPI-SKF	EnANNDeep
Precision	Dataset 1	0.8004 ± 0.0383	0.7725 ± 0.0096	**0.8638 ±0.0052**	0.8551 ± 0.0106	0.7979 ± 0.0337	0.8380 ± 0.0045
	Dataset 2	0.8101 ± 0.0162	0.7412 ± 0.0097	0.8576 ± 0.0083	**0.8688 ±0.0115**	0.7902 ± 0.0059	0.8510 ± 0.0037
	Dataset 3	0.7301 ± 0.0537	0.6793 ± 0.0082	0.7547 ± 0.0105	0.7221 ± 0.0098	**0.7631 ± 0.0095**	0.7330 ± 0.0040
	Dataset 4	0.7573 ± 0.0235	0.6167 ± 0.0101	0.6982 ± 0.0163	0.7804 ± 0.0225	**0.7948 ± 0.0070**	0.7706 ± 0.0124
	Dataset 5	0.7935 ± 0.0110	0.7002 ± 0.0123	0.7554 ± 0.0032	**0.8646 ±0.0051**	0.8248 ± 0.0011	0.8555 ± 0.0015
	Ave.	0.7783	0.7020	0.7859	**0.8182**	0.7942	0.8096
Recall	Dataset 1	0.6696 ± 0.0141	0.6858 ± 0.0086	0.8708 ± 0.0132	0.8893 ± 0.0126	0.9379 ± 0.0283	**0.9750 ± 0.0032**
	Dataset 2	0.7070 ± 0.0165	0.6514 ± 0.0104	0.8567 ± 0.0202	0.9023 ± 0.0115	0.6910 ± 0.0092	**0.9783 ± 0.0017**
	Dataset 3	0.6541 ± 0.0314	0.6141 ± 0.0093	0.6606 ± 0.0235	0.7408 ± 0.0180	0.6745 ± 0.0065	**0.8844 ± 0.0081**
	Dataset 4	0.6626 ± 0.0144	0.5601 ± 0.0083	0.6836 ± 0.0221	0.7644 ± 0.0308	0.7007 ± 0.0052	**0.8001 ± 0.0117**
	Dataset 5	0.6849 ± 0.0097	0.6307 ± 0.0075	0.8618 ± 0.0071	0.8965 ± 0.0046	0.7304 ± 0.0006	**0.9211 ± 0.0016**
	Ave.	0.6756	0.6284	0.7867	0.8433	0.7469	**0.9118**
Accuracy	Dataset 1	0.8008 ± 0.0284	0.6855 ± 0.0086	0.8666 ± 0.0047	0.8713 ± 0.0096	0.8488 ± 0.0136	**0.8932 ± 0.0027**
	Dataset 2	0.7076 ± 0.0173	0.6510 ± 0.0104	0.8571 ± 0.0071	0.8869 ± 0.0082	0.6965 ± 0.0057	**0.9034 ± 0.0023**
	Dataset 3	0.7488 ± 0.0538	0.6138 ± 0.0093	0.7217 ± 0.0036	0.7285 ± 0.0085	0.6745 ± 0.0065	**0.7810 ± 0.0035**
	Dataset 4	0.7658 ± 0.0284	0.5588 ± 0.0083	0.6930 ± 0.0127	0.7775 ± 0.0200	0.7007 ± 0.0052	**0.7805 ± 0.0100**
	Dataset 5	0.8213 ± 0.0137	0.6306 ± 0.0075	0.7913 ± 0.0022	0.8702 ± 0.0034	0.7304 ± 0.0006	**0.8828 ± 0.0009**
	Ave.	0.7689	0.6279	0.7859	0.8269	0.7302	**0.8482**
F1 score	Dataset 1	0.8189 ± 0.0159	0.6498 ± 0.0081	0.8668 ± 0.0057	0.8614 ± 0.0077	0.8742 ± 0.0094	**0.9013 ± 0.0023**
	Dataset 2	0.6740 ± 0.0158	0.6183 ± 0.0097	0.8566 ± 0.0086	0.8897 ± 0.0079	0.6565 ± 0.0071	**0.9101 ± 0.0019**
	Dataset 3	0.7767 ± 0.0251	0.5755 ± 0.0084	0.7025 ± 0.0092	0.7323 ± 0.0091	0.6359 ± 0.0072	**0.8015 ± 0.0034**
	Dataset 4	0.7745 ± 0.0227	0.5225 ± 0.0083	0.6892 ± 0.0133	0.7765 ± 0.0211	0.6636 ± 0.0057	**0.7845 ± 0.0094**
	Dataset 5	0.8351 ± 0.0104	0.5933 ± 0.0080	0.8050 ± 0.0025	0.8792 ± 0.0032	0.6923 ± 0.0007	**0.8871 ± 0.0008**
	Ave.	0.7758	0.5919	0.7840	0.8304	0.7019	**0.8569**
AUC	Dataset 1	0.8393 ± 0.0288	0.8710 ± 0.0173	0.9327 ± 0.0022	0.9265 ± 0.0069	0.9293 ± 0.0120	**0.9473 ± 0.0016**
	Dataset 2	0.9144 ± 0.0341	0.8021 ± 0.0208	0.9236 ± 0.0037	0.9385 ± 0.0058	0.8893 ± 0.0136	**0.9556 ± 0.0013**
	Dataset 3	0.8102 ± 0.0616	0.7276 ± 0.0186	0.8076 ± 0.0046	0.8071 ± 0.0108	0.8493 ± 0.0130	**0.8597 ± 0.0034**
	Dataset 4	0.8277 ± 0.0284	0.6175 ± 0.0165	0.7587 ± 0.0126	0.8506 ± 0.0186	**0.9024 ±0.0105**	0.8648 ± 0.0062
	Dataset 5	0.8672 ± 0.0180	0.7613 ± 0.0149	0.8603 ± 0.0020	0.9486 ± 0.0020	**0.9609 ± 0.0013**	0.9557 ± 0.0005
	Ave.	0.8518	0.7559	0.8566	0.8943	0.9062	**0.9166**
AUPR	Dataset 1	0.8694 ± 0.0216	0.8755 ± 0.0147	0.9186 ± 0.0028	0.8939 ± 0.0120	**0.9290 ± 0.0155**	0.9283 ± 0.0039
	Dataset 2	**0.9437 ±0.0236**	0.8310 ± 0.0134	0.9136 ± 0.0052	0.9091 ± 0.0090	0.8956 ± 0.0128	0.9408 ± 0.0018
	Dataset 3	0.8287 ± 0.0475	0.7309 ± 0.0103	**0.9136 ±0.0052**	0.7765 ± 0.0149	0.8560 ± 0.0162	0.8356 ± 0.0054
	Dataset 4	0.8209 ± 0.0187	0.6461 ± 0.0128	0.7555 ± 0.0144	0.8364 ± 0.0226	0.6683 ± 0.0061	**0.8683 ± 0.0062**
	Dataset 5	0.8527 ± 0.0114	0.7656 ± 0.0183	0.8363 ± 0.0037	0.9416 ± 0.0030	**0.9596 ± 0.0021**	0.9540 ± 0.0008
	Ave.	0.8631	0.7698	0.8675	0.8715	0.8617	**0.9054**

disease [66]. In addition, nonsmall cell lung cancer has been reported to associate with upregulated SNHG1 [67].

In the three human dataset, SNHG1 interacts with 6, 18, and 4 proteins, respectively. To find interacting proteins with SNHG1, all its interaction information is hidden. The six LPI prediction algorithms are then applied to discover potential proteins for SNHG1. The predicted top 5 proteins are shown in Table 14. It can be found that Q15717, O00425, Q9Y6M1, P35637, and Q9NZI8 are inferred to have the highest interaction probabilities with SNHG1 in dataset 1. Although interactions between the five proteins and SNHG1 are unlabeled in Dataset 1, O00425 and P35637 have been reported to have close relationships with SNHG1 in datasets 3 and 2, respectively. Q15717, Q9Y6M1, and Q9NZI8 have been shown to link with SNHG1 in both datasets 2 and 3. In addition, all the inferred top 5 proteins linking with SNHG1 have higher rankings in SFPEL, PMDKN, CatBoost, PLIPCOM, LPI-SKF, and EnANNDeep. The ranking results again demonstrate the LPI classification ability of EnANNDeep for a new lncRNA.

Table 8 Comparison of two voting methods on CV_l

Metric	Dataset	Hard voting method	Soft voting method
Precision	Dataset 1	**0.8381 ±0.0052**	0.8364 ± 0.0042
	Dataset 2	0.8516 ± 0.0050	**0.8550 ±0.0038**
	Dataset 3	0.6890 ± 0.0033	**0.6979 ±0.0056**
	Dataset 4	0.6784 ± 0.0173	**0.6882 ±0.0126**
	Dataset 5	0.7993 ± 0.0030	**0.8035 ±0.0014**
	Ave.	0.7713	**0.7762**
Recall	Dataset 1	0.9683 ± 0.0020	**0.9700 ±0.0030**
	Dataset 2	0.9662 ± 0.0025	**0.9707 ±0.0028**
	Dataset 3	**0.8140 ±0.0063**	0.8028 ± 0.0041
	Dataset 4	0.5022 ± 0.0386	**0.5628 ±0.0297**
	Dataset 5	0.8599 ± 0.0020	**0.8616 ±0.0024**
	Ave.	0.8221	**0.8336**
Accuracy	Dataset 1	**0.8904 ±0.0034**	0.8900 ± 0.0033
	Dataset 2	0.8988 ± 0.0036	**0.9029 ±0.0031**
	Dataset 3	0.7231 ± 0.0025	**0.7275 ±0.0045**
	Dataset 4	0.6315 ± 0.0176	**0.6530 ±0.0122**
	Dataset 5	0.8218 ± 0.0020	**0.8253 ±0.0007**
	Ave.	0.7931	**0.7997**
F1 score	Dataset 1	**0.8984 ±0.0028**	0.8982 ± 0.0029
	Dataset 2	0.9051 ± 0.0031	**0.9091 ±0.0029**
	Dataset 3	0.7461 ± 0.0024	**0.7465 ±0.0033**
	Dataset 4	0.5727 ± 0.0312	**0.6149 ±0.0202**
	Dataset 5	0.8283 ± 0.0017	**0.8314 ±0.0008**
	Ave.	0.7901	**0.8000**
AUC	Dataset 1	**0.9347 ±0.0025**	0.9330 ± 0.0023
	Dataset 2	0.9479 ± 0.0030	**0.9487 ±0.0021**
	Dataset 3	0.8188 ± 0.0028	**0.8199 ±0.0030**
	Dataset 4	**0.7234 ±0.0145**	0.7189 ± 0.0104
	Dataset 5	**0.9097 ±0.0010**	0.9093 ± 0.0008
	Ave.	**0.8669**	0.8660
AUPR	Dataset 1	**0.9183 ±0.0035**	0.9154 ± 0.0043
	Dataset 2	0.9342 ± 0.0043	**0.9350 ±0.0030**
	Dataset 3	0.8010 ± 0.0045	**0.8014 ±0.0040**
	Dataset 4	**0.7170 ±0.0122**	0.7133 ± 0.0092
	Dataset 5	**0.9080 ±0.0013**	0.9073 ± 0.0012
	Ave.	**0.8577**	0.8545

Table 9 The performance of two voting methods on CV_p

Metric	Dataset	Hard voting method	Soft voting method
Precision	Dataset 1	0.8343 ± 0.1003	**0.8758 ±0.0547**
	Dataset 2	0.8480 ± 0.0310	**0.8885 ±0.037**2
	Dataset 3	0.7255 ± 0.0340	**0.7420 ±0.031**3
	Dataset 4	0.7526 ± 0.0130	**0.7667 ±0.0133**
	Dataset 5	0.8364 ± 0.0032	**0.8381 ±0.0053**
	Ave.	0.7994	**0.8222**
Recall	Dataset 1	0.4069 ± 0.0600	**0.4451 ±0.1061**
	Dataset 2	0.2995 ± 0.0528	**0.3573 ±0.0564**
	Dataset 3	0.5876 ± 0.0687	**0.6774 ±0.0426**
	Dataset 4	0.7373 ± 0.0202	**0.7495 ±0.0233**
	Dataset 5	**0.8582 ±0.0030**	0.8521 ± 0.0063
	Ave.	0.5779	**0.6163**
Accuracy	Dataset 1	0.6754 ± 0.0264	**0.6968 ±0.0505**
	Dataset 2	0.6260 ± 0.0254	**0.6580 ±0.0279**
	Dataset 3	0.6776 ± 0.0262	**0.7147 ±0.0267**
	Dataset 4	0.7451 ± 0.0112	**0.7595 ±0.0121**
	Dataset 5	**0.8445 ±0.0016**	0.8433 ± 0.0029
	Ave.	0.7137	**0.7345**
F1 score	Dataset 1	0.4917 ± 0.0624	**0.5436 ±0.1086**
	Dataset 2	0.4031 ± 0.0639	**0.4656 ±0.0661**
	Dataset 3	0.6342 ± 0.0460	**0.6965 ±0.0274**
	Dataset 4	0.7400 ± 0.0129	**0.7554 ±0.0153**
	Dataset 5	**0.8467 ±0.0014**	0.8446 ± 0.0030
	Ave.	0.6231	**0.6611**
AUC	Dataset 1	0.9061 ± 0.0282	**0.9093 ±0.0106**
	Dataset 2	0.9125 ± 0.0193	**0.9175 ±0.0134**
	Dataset 3	**0.7944 ±0.0208**	0.7907 ± 0.0247
	Dataset 4	0.8357 ± 0.0127	**0.8422 ±0.0090**
	Dataset 5	0.9269 ± 0.0015	**0.9280 ±0.0018**
	Ave.	0.8751	**0.8775**
AUPR	Dataset 1	0.8752 ± 0.0291	**0.8775 ±0.0150**
	Dataset 2	0.8721 ± 0.0226	**0.8803 ±0.0193**
	Dataset 3	**0.7767 ±0.0229**	0.7732 ± 0.0251
	Dataset 4	0.8389 ± 0.0130	**0.8444 ±0.0118**
	Dataset 5	0.9196 ± 0.0019	**0.9221 ±0.0022**
	Ave.	0.8565	**0.8595**

3.6.2 Finding Interacting lncRNAs for New Proteins

Q9UKV8 can inhibit translation initiation through interaction with translation initiation factor EIF6 and prevent the recruitment from translation initiation factor EIF4-E. It upregulates translation under the situation of serum starvation by binding to the AU element. More importantly, it is also interrelated with transcriptional gene silencing [41].

Q9UKV8 interacts with 207, 205, and 222 lncRNAs on three human datasets, respectively. In this section, its association information with lncRNAs is hidden and EnANNDeep is used to reveal its relevant lncRNAs. The found top 5 human lncRNAs interacting with Q9UKV8 are shown in Table 15.

On dataset 1, it can be observed that DANCR, RPI001_1039837 and AL139819.1 are inferred to interact with Q9UKV8. Although the associations between the three lncRNAs and Q9UKV8 are unknown in dataset 1, DANCR has been reported to interact with Q9UKV8 in dataset 2, RPI001_1039837 and AL139819.1 have been validated to interact with Q9UKV8 in dataset 3.

Table 10 The performance of two voting methods on CV_{lp}

Metric	Dataset	Hard voting method	Soft voting method
Precision	Dataset 1	0.8373 ± 0.0024	**0.8380 ±0.0045**
	Dataset 2	**0.8515 ±0.0024**	0.8510 ± 0.0037
	Dataset 3	0.7168 ± 0.0048	**0.7330 ±0.0040**
	Dataset 4	**0.7790 ±0.0122**	0.7706 ± 0.0124
	Dataset 5	0.8391 ± 0.0011	**0.8555 ±0.0015**
	Ave.	0.8047	**0.8096**
Recall	Dataset 1	0.9722 ± 0.0026	**0.9750 ±0.0032**
	Dataset 2	**0.9784 ±0.0024**	0.9783 ± 0.0017
	Dataset 3	0.8780 ± 0.0027	**0.8844 ±0.0081**
	Dataset 4	**0.8192 ±0.0111**	0.8001 ± 0.0117
	Dataset 5	0.9206 ± 0.0008	**0.9211 ±0.0016**
	Ave.	**0.9137**	0.9118
Accuracy	Dataset 1	0.8915 ± 0.0024	**0.8932 ±0.0027**
	Dataset 2	**0.9037 ±0.0019**	0.9034 ± 0.0023
	Dataset 3	0.7654 ± 0.0038	**0.7810 ±0.0035**
	Dataset 4	**0.7928 ±0.0101**	0.7805 ± 0.0100
	Dataset 5	0.8720 ± 0.0007	**0.8828 ±0.0009**
	Ave.	0.8451	**0.8482**
F1 score	Dataset 1	0.8997 ± 0.0023	**0.9013 ±0.0023**
	Dataset 2	**0.9105 ±0.0017**	0.9101 ± 0.0019
	Dataset 3	0.7892 ± 0.0027	**0.8015 ±0.0034**
	Dataset 4	**0.7979 ±0.0097**	0.7845 ± 0.0094
	Dataset 5	0.8779 ± 0.0006	**0.8871 ±0.0008**
	Ave.	0.8550	**0.8569**
AUC	Dataset 1	0.9456 ± 0.0018	**0.9473 ±0.0016**
	Dataset 2	**0.9573 ±0.0026**	0.9556 ± 0.0013
	Dataset 3	0.8574 ± 0.0018	**0.8597 ±0.0034**
	Dataset 4	0.8636 ± 0.0069	**0.8648 ±0.0062**
	Dataset 5	0.9557 ± 0.0006	**0.9557 ±0.0005**
	Ave.	0.9159	**0.9166**
AUPR	Dataset 1	0.9261 ± 0.0036	**0.9283 ±0.0039**
	Dataset 2	**0.9427 ±0.0042**	0.9408 ± 0.0018
	Dataset 3	0.8350 ± 0.0044	**0.8356 ±0.0054**
	Dataset 4	0.8678 ± 0.0077	**0.8683 ±0.0062**
	Dataset 5	0.9540 ± 0.0008	**0.9540 ±0.0008**
	Ave.	0.9051	**0.9054**

On dataset 2, it can be seen that RMRP, SNORD17, and RPI001_483534 have been predicted to interact with Q9UKV8. Although the relationships between RMRP and SNORD17 and Q9UKV8 are unknown in dataset 2, the two lncRNAs have been shown to link with Q9UKV8 in datasets 1 and 3, respectively. The interaction between RPI001_483534 and Q9UKV8 can not be retrieved on the three datasets. However, it ranks as 5, 2, 478, 183, 9, and 86 by EnANNDeep, SPFEL, PMDKN, PLIPCOM, LPI-CatBoost, and LPI-SKF, respectively. The higher rankings in EnANNDeep, SPFEL, and PLIPCOM suggest that RPI001_483534 may be relative to Q9UKV8 and needs further validation.

On dataset 3, RPI001_84645 and EXOC3 are identified to interact with Q9UKV8. The associations between the two lncRNAs and Q9UKV8 can be searched in datasets 2.

3.6.3 Finding New LPIs Based on Known LPIs

Potential LPIs are subsequently identified by EnANNDeep based on labeled LPIs. The inferred top 50 LPIs with the highest correlation probabilities on the five datasets are illustrated in Figs. 4, 5, 6, 7 and 8. The 50 associations contain known and unknown LPIs.

The ranking results show that interactions between SNHG10 and Q15717, VIM-AS1, and Q15717, RPI001_101_2148 and ENSP00000385269, AthlncRNA-159 and 22328551, and ZmalncRNA-1314 and B6SP74 are the most possible LPIs among unlabeled lncRNA–protein pairs on datasets 1–5, respectively. They are ranked as 4, 14, 5, 33, and 1972 among 55,165, 74,340, 26,730, 3815, and 71,568 lncRNA–protein pairs, respectively.

lncRNA SNHG10 is a novel driver in the process of development and metastasis in hepatocellular carcinoma [68]. The lncRNA has close linkages with cell proliferation in gastric cancer [69], non-small cell lung cancer [70], and osteosarcoma [71]. Q15717 is an RNA-binding protein [72]. The protein contributes to embryonic stem cell differentiation, and can increase the leptin mRNA's stability, and mediate the CDKN2A anti-proliferative activity [41]. Both SNHG10 and Q15717 have dense linkages with cell proliferation activity. We infer that SNHG10 may interact with Q15717 and is worthy of further validation.

4 Discussion

Identification of LPI candidates contributes to discover functions and mechanisms of lncRNAs. In this manuscript, an ensemble framework combining A*k*NN, DNN, and deep forest is developed to find possible interactions between lncRNAs and proteins. Three different CVs are conducted to compare the proposed EnANNDeep model with the other LPI prediction methods. The experimental results indicate that EnANNDeep can be more accurately applied to new LPI discovery.

Under CV_p, majority of performance achieved from SFPEL, PMDKN, CatBoost, PLIPCOM, and LPI-SKF is much lower than those of CV_l and CV_{lp}. Under CV_p, 80% of lncRNAs are used to train the model and the remaining is applied to test the model. On five LPI datasets, each lncRNA may associate with 59, 84, 27, 35, and 42 proteins, respectively. When 20% of proteins are masked their associations, it may shield many LPIs and thus reduce the fitting level

Table 11 The effect of the number of RNA secondary structures on performance under CV_l

Metric	Dataset	128	64	10	5
Precision	Dataset 1	0.8357 ± 0.0027	0.8388 ± 0.0056	0.8393 ± 0.0053	0.8381 ± 0.0052
	Dataset 2	0.8528 ± 0.0058	0.8545 ± 0.0032	0.8533 ± 0.0044	0.8516 ± 0.0050
	Dataset 3	0.6919 ± 0.0060	0.6880 ± 0.0033	0.6923 ± 0.0030	0.6890 ± 0.0033
	Dataset 4	0.6752 ± 0.0192	0.6623 ± 0.0148	0.6657 ± 0.0213	0.6784 ± 0.0173
	Dataset 5	0.8008 ± 0.0026	0.7995 ± 0.0031	0.7990 ± 0.0024	0.7993 ± 0.0030
	Ave.	0.7713	0.7686	0.7699	0.7713
Recall	Dataset 1	0.9680 ± 0.0029	0.9660 ± 0.0023	0.9679 ± 0.0033	0.9683 ± 0.0020
	Dataset 2	0.9665 ± 0.0032	0.9618 ± 0.0028	0.9649 ± 0.0030	0.9662 ± 0.0025
	Dataset 3	0.8033 ± 0.0068	0.8220 ± 0.0062	0.8128 ± 0.0063	0.8140 ± 0.0063
	Dataset 4	0.5030 ± 0.0306	0.4848 ± 0.0270	0.4821 ± 0.0428	0.5022 ± 0.0386
	Dataset 5	0.8619 ± 0.0039	0.8587 ± 0.0035	0.8585 ± 0.0022	0.8599 ± 0.0020
	Ave.	0.8205	0.8187	0.8172	0.8221
Accuracy	Dataset 1	0.8887 ± 0.0020	0.8901 ± 0.0042	0.8912 ± 0.0041	0.8904 ± 0.0034
	Dataset 2	0.9001 ± 0.0037	0.8990 ± 0.0025	0.8994 ± 0.0032	0.8988 ± 0.0036
	Dataset 3	0.7241 ± 0.0050	0.7245 ± 0.0036	0.7255 ± 0.0026	0.7231 ± 0.0025
	Dataset 4	0.6202 ± 0.0149	0.6182 ± 0.0119	0.6188 ± 0.0168	0.6315 ± 0.0176
	Dataset 5	0.8243 ± 0.0013	0.8216 ± 0.0023	0.8211 ± 0.0024	0.8218 ± 0.0020
	Ave.	0.7915	0.7907	0.7912	0.7931
F1 score	Dataset 1	0.8969 ± 0.0018	0.8979 ± 0.0036	0.8990 ± 0.0036	0.8984 ± 0.0028
	Dataset 2	0.9059 ± 0.0032	0.9049 ± 0.0023	0.9056 ± 0.0029	0.9051 ± 0.0031
	Dataset 3	0.7450 ± 0.0040	0.7489 ± 0.0036	0.7475 ± 0.0027	0.7461 ± 0.0024
	Dataset 4	0.5726 ± 0.0203	0.5552 ± 0.0195	0.5548 ± 0.0316	0.5727 ± 0.0312
	Dataset 5	0.8306 ± 0.0014	0.8280 ± 0.0022	0.8276 ± 0.0015	0.8283 ± 0.0017
	Ave.	0.7902	0.7870	0.7869	0.7901
AUC	Dataset 1	0.9329 ± 0.0021	0.9327 ± 0.0030	0.9341 ± 0.0021	0.9347 ± 0.0025
	Dataset 2	0.9473 ± 0.0023	0.9475 ± 0.0017	0.9469 ± 0.0022	0.9479 ± 0.0030
	Dataset 3	0.8188 ± 0.0044	0.9469 ± 0.0022	0.8189 ± 0.0023	0.8188 ± 0.0028
	Dataset 4	0.7188 ± 0.0160	0.7154 ± 0.0079	0.7136 ± 0.0142	0.7234 ± 0.0145
	Dataset 5	0.9087 ± 0.0012	0.9091 ± 0.0017	0.9089 ± 0.0015	0.9097 ± 0.0010
	Ave.	0.8653	0.8648	0.8645	0.8669
AUPR	Dataset 1	0.9148 ± 0.0041	0.9142 ± 0.0038	0.9159 ± 0.0039	0.9183 ± 0.0035
	Dataset 2	0.9329 ± 0.0044	0.9338 ± 0.0026	0.9325 ± 0.0033	0.9342 ± 0.0043
	Dataset 3	0.8000 ± 0.0054	0.7990 ± 0.0044	0.7995 ± 0.0022	0.8010 ± 0.0045
	Dataset 4	0.7135 ± 0.0147	0.7090 ± 0.0108	0.7071 ± 0.0131	0.7170 ± 0.0122
	Dataset 5	0.9073 ± 0.0014	0.9072 ± 0.0018	0.9065 ± 0.0017	0.9080 ± 0.0013
	Ave.	0.8537	0.8526	0.8523	0.8557

of a classification model to the LPI data. Therefore, abundance level of data severely affects the learning capacity of the five models. In comparison, under CV_p, the performance obtained from EnANNDeep keeps relatively steady or even outperforms ones in comparing to CV_l and CV_{lp}. The results demonstrate the robustness of the proposed EnANNDeep algorithm under CVs.

More importantly, similar to EnANNDeep, SFPEL, CatBoost, and PLIPCOM are three ensemble learning-based algorithms. The four ensemble learning-based LPI prediction methods integrate sequence information related to lncRNAs and proteins. SFPEL is a feature projection-based technique. CatBoost and PLIPCOM are gradient tree boosting and categorical boosting algorithms, respectively. EnANNDeep outperforms the three ensemble learning models, demonstrating the superior classification ability of basic predictors. That is, A*k*NN, DNN, and deep forest can be more effectively integrated to find possible LPIs. In addition, a few case analysis further suggest that EnANNDeep can mine useful information for a new lncRNA or protein.

Table 12 The effect of the number of RNA secondary structures on performance under CV_p

Metric	Dataset	128	64	10	5
Precision	Dataset 1	0.8376 ± 0.0728	0.8368 ± 0.0955	0.8529 ± 0.0806	0.8343 ± 0.1003
	Dataset 2	0.8853 ± 0.0201	0.8821 ± 0.0201	0.8816 ± 0.0284	0.8480 ± 0.0310
	Dataset 3	0.7275 ± 0.0166	0.7039 ± 0.0244	0.6729 ± 0.0245	0.7255 ± 0.0340
	Dataset 4	0.7581 ± 0.0166	0.7527 ± 0.0115	0.7566 ± 0.0152	0.7526 ± 0.0130
	Dataset 5	0.8212 ± 0.0041	0.8183 ± 0.0047	0.8256 ± 0.0032	0.8364 ± 0.0032
	Ave.	0.8059	0.7988	0.7979	0.7994
Recall	Dataset 1	0.3072 ± 0.0825	0.2897 ± 0.1123	0.3668 ± 0.0573	0.4069 ± 0.0600
	Dataset 2	0.2415 ± 0.0698	0.3698 ± 0.0586	0.4136 ± 0.0549	0.2995 ± 0.0528
	Dataset 3	0.5811 ± 0.0363	0.5778 ± 0.0401	0.5828 ± 0.0371	0.5876 ± 0.0687
	Dataset 4	0.7495 ± 0.0206	0.7552 ± 0.0244	0.7618 ± 0.0202	0.7373 ± 0.0202
	Dataset 5	0.8546 ± 0.0041	0.8874 ± 0.0056	0.8696 ± 0.0045	0.8582 ± 0.0030
	Ave.	0.5468	0.5760	0.5989	0.5779
Accuracy	Dataset 1	0.6222 ± 0.0387	0.6258 ± 0.0521	0.6609 ± 0.0271	0.6754 ± 0.0264
	Dataset 2	0.6046 ± 0.0325	0.6632 ± 0.0285	0.6818 ± 0.0300	0.6260 ± 0.0254
	Dataset 3	0.6610 ± 0.0154	0.6626 ± 0.0223	0.6485 ± 0.0196	0.6776 ± 0.0262
	Dataset 4	0.7500 ± 0.0118	0.7514 ± 0.0086	0.7567 ± 0.0112	0.7451 ± 0.0112
	Dataset 5	0.8471 ± 0.0026	0.8447 ± 0.0024	0.8424 ± 0.0025	0.8445 ± 0.0016
	Ave.	0.6970	0.7095	0.7181	0.7137
F1 score	Dataset 1	0.4095 ± 0.0837	0.3745 ± 0.1170	0.4555 ± 0.0658	0.4917 ± 0.0624
	Dataset 2	0.3442 ± 0.0783	0.4839 ± 0.0588	0.5275 ± 0.0526	0.4031 ± 0.0639
	Dataset 3	0.6159 ± 0.0265	0.6145 ± 0.0316	0.6128 ± 0.0274	0.6342 ± 0.0460
	Dataset 4	0.7413 ± 0.0117	0.7502 ± 0.0118	0.7562 ± 0.0121	0.7400 ± 0.0129
	Dataset 5	0.8450 ± 0.0022	0.8510 ± 0.0022	0.8466 ± 0.0025	0.8467 ± 0.0014
	Ave.	0.5912	0.6148	0.6397	0.6231
AUC	Dataset 1	0.9054 ± 0.0304	0.8886 ± 0.0450	0.9011 ± 0.0347	0.9061 ± 0.0282
	Dataset 2	0.9169 ± 0.0143	0.9270 ± 0.0060	0.9248 ± 0.0104	0.9125 ± 0.0193
	Dataset 3	0.7900 ± 0.0097	0.7930 ± 0.0205	0.7726 ± 0.0171	0.7944 ± 0.0208
	Dataset 4	0.8410 ± 0.0115	0.8436 ± 0.0072	0.8496 ± 0.0097	0.8357 ± 0.0127
	Dataset 5	0.9279 ± 0.0018	0.9345 ± 0.0018	0.9267 ± 0.0016	0.9269 ± 0.0015
	Ave.	0.8762	0.8773	0.8750	0.8751
AUPR	Dataset 1	0.8719 ± 0.0325	0.8540 ± 0.0467	0.8672 ± 0.0310	0.8752 ± 0.0291
	Dataset 2	0.8743 ± 0.0163	0.8910 ± 0.0093	0.8913 ± 0.0142	0.8721 ± 0.0226
	Dataset 3	0.7715 ± 0.0131	0.7714 ± 0.0204	0.7546 ± 0.0183	0.7767 ± 0.0229
	Dataset 4	0.8414 ± 0.0150	0.8474 ± 0.0079	0.8494 ± 0.0111	0.8389 ± 0.0130
	Dataset 5	0.9226 ± 0.0022	0.9292 ± 0.0024	0.9208 ± 0.0019	0.9196 ± 0.0019
	Ave.	0.8563	0.8586	0.8567	0.8565

The EnANNDeep framework demonstrates the powerful LPI discovery ability, especially under CV_{lp}. It may be attributed to the following characteristics. First, a deep model composed of DNN and deep forest exhibits the optimal feature representation ability. In particular, deep forest works well even on small-scale data. Second, the proposed A*k*NN classifier can separately pick the most appropriate *k* for each query point so that the algorithm can better set the confidence intervals. Third, the ensemble framework from A*k*NN, DNN, and deep forest can effectively integrate the prediction results from the three predictors and thus improves the classification performance of EnANNDeep. Finally, it integrates multiple biological information related to LPI.

Although EnANNDeep can precisely identify new LPIs, it has one limitation: we select negative LPIs from unlabeled lncRNA–protein pairs. Indeed, unknown lncRNA–protein pairs may contain positive LPIs, thereby affecting the prediction ability of a model.

Table 13 The effect of the number of RNA secondary structures on performance under CV_{lp}

Metric	Dataset	128	64	10	5
Precision	Dataset 1	0.8373 ± 0.0050	0.8410 ± 0.0046	0.8399 ± 0.0042	0.8373 ± 0.0024
	Dataset 2	0.8514 ± 0.0046	0.8528 ± 0.0039	0.8549 ± 0.0038	0.8515 ± 0.0024
	Dataset 3	0.7149 ± 0.0054	0.7100 ± 0.0077	0.7102 ± 0.0042	0.7168 ± 0.0048
	Dataset 4	0.7773 ± 0.0093	0.7737 ± 0.0110	0.7769 ± 0.0087	0.7790 ± 0.0122
	Dataset 5	0.8442 ± 0.0025	0.8406 ± 0.0021	0.8397 ± 0.0011	0.8391 ± 0.0011
	Ave.	0.8050	0.8036	0.8043	0.8153
Recall	Dataset 1	0.9737 ± 0.0026	0.9697 ± 0.0025	0.9710 ± 0.0034	0.9722 ± 0.0026
	Dataset 2	0.9772 ± 0.0018	0.9775 ± 0.0025	0.9777 ± 0.0019	0.9784 ± 0.0024
	Dataset 3	0.8729 ± 0.0075	0.8765 ± 0.0122	0.8767 ± 0.0118	0.8780 ± 0.0027
	Dataset 4	0.8065 ± 0.0121	0.8230 ± 0.0077	0.8198 ± 0.0095	0.8192 ± 0.0111
	Dataset 5	0.9241 ± 0.0016	0.9207 ± 0.0025	0.9185 ± 0.0015	0.9206 ± 0.0008
	Ave.	0.9109	0.9135	0.9127	0.9137
Accuracy	Dataset 1	0.8921 ± 0.0031	0.8931 ± 0.0033	0.8929 ± 0.0026	0.8915 ± 0.0024
	Dataset 2	0.9036 ± 0.0031	0.9043 ± 0.0024	0.9058 ± 0.0027	0.9037 ± 0.0019
	Dataset 3	0.7638 ± 0.0048	0.7589 ± 0.0046	0.7592 ± 0.0030	0.7654 ± 0.0038
	Dataset 4	0.7874 ± 0.0086	0.7903 ± 0.0072	0.7904 ± 0.0059	0.7928 ± 0.0101
	Dataset 5	0.8731 ± 0.0014	0.8730 ± 0.0012	0.8715 ± 0.0009	0.8720 ± 0.0007
	Ave.	0.8440	0.8439	0.8440	0.8451
F1 score	Dataset 1	0.9003 ± 0.0026	0.9007 ± 0.0028	0.9007 ± 0.0022	0.8997 ± 0.0023
	Dataset 2	0.9101 ± 0.0026	0.9109 ± 0.0021	0.9121 ± 0.0023	0.9105 ± 0.0017
	Dataset 3	0.7860 ± 0.0041	0.7843 ± 0.0033	0.7845 ± 0.0037	0.7892 ± 0.0027
	Dataset 4	0.7912 ± 0.0086	0.7968 ± 0.0059	0.7949 ± 0.0054	0.7979 ± 0.0097
	Dataset 5	0.8777 ± 0.0012	0.8788 ± 0.0011	0.8773 ± 0.0009	0.8779 ± 0.0006
	Ave.	0.8531	0.8543	0.8539	0.8550
AUC	Dataset 1	0.9455 ± 0.0028	0.9454 ± 0.0019	0.9470 ± 0.0019	0.9456 ± 0.0018
	Dataset 2	0.9563 ± 0.0023	0.9579 ± 0.0018	0.9567 ± 0.0022	0.9573 ± 0.0026
	Dataset 3	0.8551 ± 0.0028	0.8583 ± 0.0026	0.8569 ± 0.0019	0.8574 ± 0.0018
	Dataset 4	0.8602 ± 0.0067	0.8629 ± 0.0054	0.8630 ± 0.0089	0.8636 ± 0.0069
	Dataset 5	0.9561 ± 0.0007	0.9560 ± 0.0003	0.9553 ± 0.0004	0.9557 ± 0.0006
	Ave.	0.9146	0.9161	0.9158	0.9159
AUPR	Dataset 1	0.9250 ± 0.0057	0.9246 ± 0.0040	0.9277 ± 0.0036	0.9261 ± 0.0036
	Dataset 2	0.9411 ± 0.0031	0.9435 ± 0.0034	0.9422 ± 0.0036	0.9427 ± 0.0042
	Dataset 3	0.8314 ± 0.0041	0.8353 ± 0.0038	0.8338 ± 0.0043	0.8350 ± 0.0044
	Dataset 4	0.8731 ± 0.0074	0.8715 ± 0.0067	0.8648 ± 0.0091	0.8678 ± 0.0077
	Dataset 5	0.9543 ± 0.0008	0.9546 ± 0.0004	0.9537 ± 0.0005	0.9540 ± 0.0008
	Ave.	0.9050	0.9059	0.9044	0.9051

5 Conclusions

lncRNAs play pivotal roles in regulating many hallmarks of cancer biology. To decipher the lncRNA functions, we focus on new LPI mining. First, five LPI-related datasets are arranged. Second, the lncRNA and protein features are fused to depict each lncRNA–protein pair. Third, an ensemble model, composed of A*k*NN, DNN, and deep forest, is developed to classify unlabeled lncRNA–protein pairs, respectively. Finally, interaction probabilities of each lncRNA–protein pair from three predictors are integrated based on a soft voting technique to obtain the final classification. The results from comparative experiments and

Table 14 The predicted top 5 proteins interacting with SNHG1

Dataset	Proteins	Confirmed	EnANNDeep	SPFEL	PMDKN	PLIPCOM	CatBoost	LPI-SKF
Dataset 1	Q15717	NO	1	1	1	3	1	7
	O00425	NO	2	2	2	10	3	8
	Q9Y6M1	NO	3	3	4	1	2	9
	P35637	NO	4	5	59	2	5	11
	Q9NZI8	NO	5	4	3	12	4	10
Dataset 2	Q15717	YES	1	1	2	2	2	6
	Q9Y6M1	YES	2	2	1	9	1	8
	Q9NZI8	YES	3	3	16	3	3	4
	Q13148	YES	4	6	11	4	8	10
	P35637	YES	5	4	20	1	4	7
Dataset 3	Q15717	YES	1	1	1	1	1	4
	Q9Y6M1	YES	2	3	5	5	4	1
	O00425	YES	3	2	6	2	2	3
	Q9NZI8	YES	4	4	4	3	3	2
	P35637	NO	5	5	8	11	5	5

Table 15 The predicted top 5 lncRNAs interacting with Q9UKV8

Dataset	lncRNAs	Confirmed	EnANNDeep	SPFEL	PMDKN	PLIPCOM	CatBoost	LPI-SKF
Dataset 1	RPI001_448664	YES	1	157	58	262	138	543
	DANCR	NO	2	145	132	1	7	114
	RPI001_1039837	NO	3	59	196	5	2	123
	RPI001_124004	YES	4	307	46	236	27	35
	AL139819.1	NO	5	538	129	129	114	178
Dataset 2	RMRP	NO	1	14	305	8	6	69
	SNORA53	YES	2	53	59	177	15	638
	RPI001_84645	YES	3	313	864	59	54	317
	SNORD17	NO	4	401	785	22	17	205
	RPI001_483534	NO	5	2	478	183	9	86
Dataset 3	RMRP	YES	1	63	196	79	92	30
	AC010890.1	YES	2	7	115	1	3	109
	RPI001_1001088	YES	3	24	15	10	2	217
	RPI001_84645	NO	4	86	888	62	52	185
	EXOC3	NO	5	604	276	297	140	841

case analyses demonstrate that EnANNDeep can optimize the interplays between lncRNAs and proteins. Case analyses suggest that there probably exists an interaction between SNHG10 and Q15717.

In the future researches, first we will integrate various lncRNA-related datasets from different data sources to investigate the interaction biomolecules for lncRNAs, for example, lncRNA-miRNA interactions [73] and lncRNA-DNA interactions [36]. Second, more biological information from lncRNAs and proteins, for example, secondary structures of lncRNAs, secondary and tertiary structures of proteins, will be fused to represent an lncRNA–protein pair. Finally, we will develop a negative sample selection method

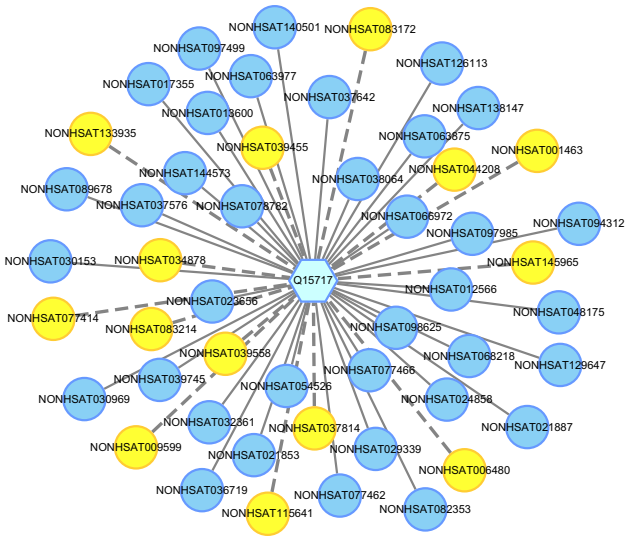


Fig. 4 The predicted top 50 LPIs on dataset 1

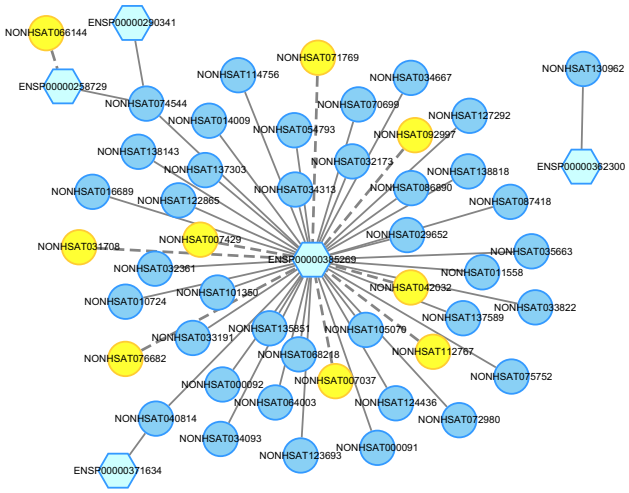


Fig. 6 The predicted top 50 LPIs on dataset 3

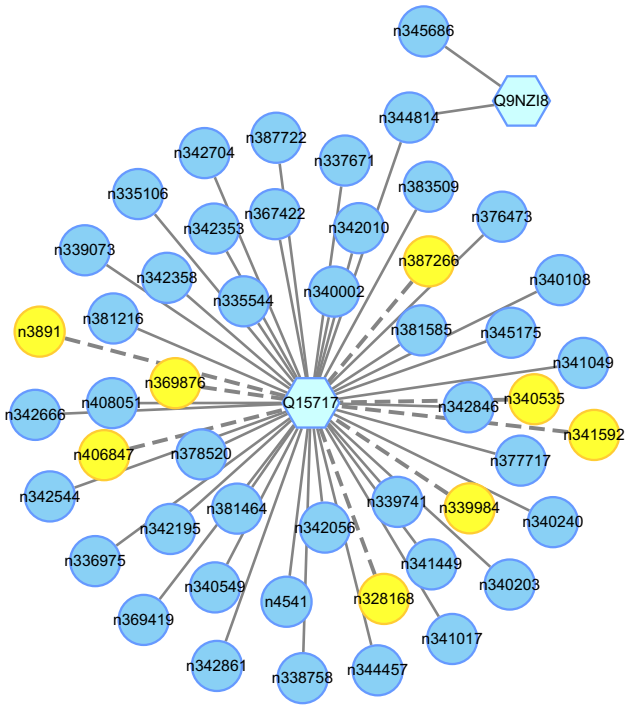


Fig. 5 The predicted top 50 LPIs on dataset 2

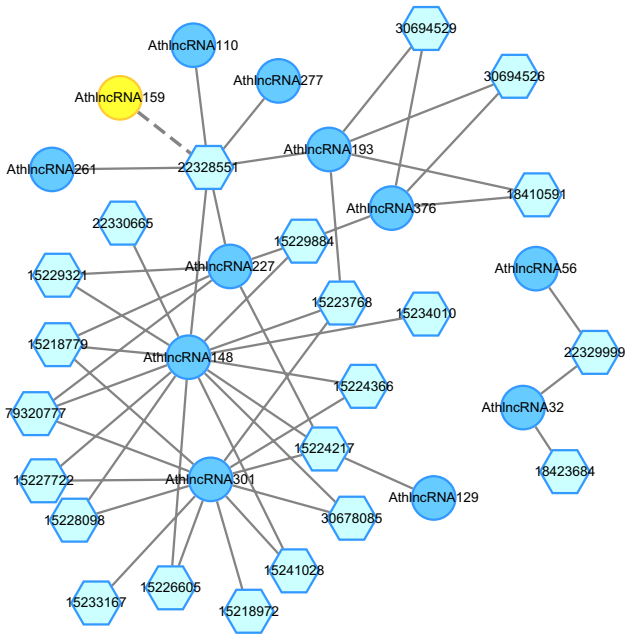


Fig. 7 The predicted top 50 LPIs on dataset 4

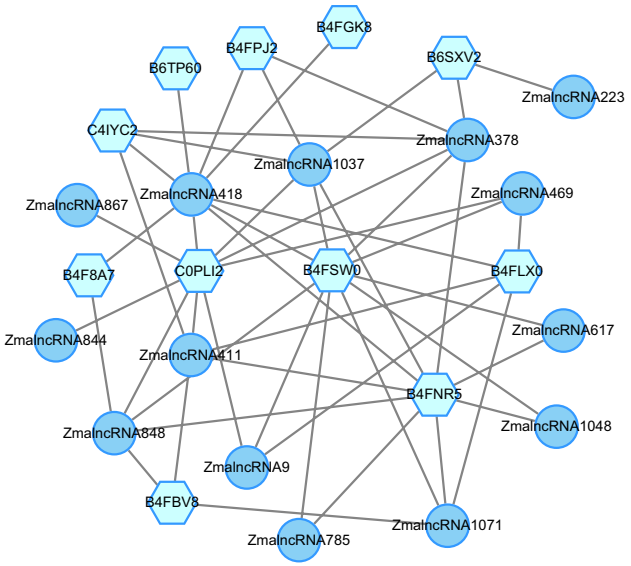


Fig. 8 The predicted top 50 LPIs on dataset 5

based on positive-unlabeled learning to screen reliable negative LPIs.

Acknowledgements We would like to thank anonymous reviewers and all authors of the cited references.

Author Contributions Conceptualization: L-HP, J-WT and L-QZ; funding acquisition: L-HP, L-QZ; investigation: L-HP and J-WT; methodology: L-HP and J-WT; project administration: L-HP, L-QZ; software: J-WT; validation: J-WT, X-FT; writing—original draft: L-HP; writing—review and editing: L-HP and J-WT.

Funding This research was funded by the National Natural Science-Foundation of China (Grant 62072172, 61803151).

Availability of data and materials Source codes and datasets are freely available for download at https://github.com/plhhnu/EnANNDeep.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Code availability Source codes are freely available for download at https://github.com/plhhnu/EnANNDeep.

References

1. Chen X, Sun YZ, Guan NN, Qu J, Huang ZA, Zhu ZX, Li JQ (2019) Computational models for LNCRNA function prediction and functional similarity calculation. Brief Funct Genom 18(1):58–82. https://doi.org/10.1093/bfgp/ely031
2. Wang J, Ma R, Ma W, Chen J, Yang J, Xi Y, Cui Q (2016) Lncdisease: a sequence based bioinformatics tool for predicting lncRNA-disease associations. Nucleic Acids Res 44(9):e90–e90. https://doi.org/10.1093/nar/gkw093
3. Ching T, Masaki J, Weirather J, Garmire LX (2015) Non-coding yet non-trivial: a review on the computational genomics of lincrnas. BioData Min 8(1):1–12. https://doi.org/10.1186/s13040-015-0075-z
4. Zhang H, Ming Z, Fan C, Zhao Q, Liu H (2020) A path-based computational model for long non-coding RNA-protein interaction prediction. Genomics 112(2):1754–1760. https://doi.org/10.1016/j.ygeno.2019.09.018
5. Chen X, Yan CC, Zhang X, You ZH (2017) Long non-coding RNAS and complex diseases: from experimental results to computational models. Brief Bioinform 18(4):558–576. https://doi.org/10.1093/bib/bbw060
6. Wang W, Dai Q, Li F, Xiong Y, Wei DQ (2020) Mlcdforest: multi-label classification with deep forest in disease prediction for long non-coding rnas. Brief Bioinform. https://doi.org/10.1093/bib/bbaa104
7. Liu H, Song G, Zhou L, Hu X, Liu M, Nie J, Lu S, Wu X, Cao Y, Tao L et al (2013) Compared analysis of LNCRNA expression profiling in pdk1 gene knockout mice at two time points. Cell Physiol Biochem 32(5):1497–1508. https://doi.org/10.1159/000356586
8. Vizoso M, Esteller M (2012) The activatory long non-coding RNA dbe-t reveals the epigenetic etiology of facioscapulohumeral muscular dystrophy. Cell Res 22(10):1413–1415. https://doi.org/10.1038/cr.2012.93
9. De R, Hu T, Moore JH, Gilbert-Diamond D (2015) Characterizing gene-gene interactions in a statistical epistasis network of twelve candidate genes for obesity. BioData Min 8(1):1–16. https://doi.org/10.1186/s13040-015-0077-x
10. Wang J, Su Z, Lu S, Fu W, Liu Z, Jiang X, Tai S (2018) Lncrna hoxa-as2 and its molecular mechanisms in human cancer. Clin Chim Acta 485:229–233. https://doi.org/10.1016/j.cca.2018.07.004
11. Tamang S, Acharya V, Roy D, Sharma R, Aryaa A, Sharma U, Khandelwal A, Prakash H, Vasquez KM, Jain A (2019) Snhg12: an lncRNA as a potential therapeutic target and biomarker for human cancer. Front Oncol 9:901. https://doi.org/10.3389/fonc.2019.00901
12. Liu T, Han Z, Li H, Zhu Y, Sun Z, Zhu A (2018) Lncrna dleu1 contributes to colorectal cancer progression via activation of kpna3. Mol Cancer 17(1):1–13. https://doi.org/10.1186/s12943-018-0873-2
13. Loewen G, Jayawickramarajah J, Zhuo Y, Shan B (2014) Functions of LNCRNA hotair in lung cancer. J Hematol Oncol 7(1):1–10. https://doi.org/10.1186/s13045-014-0090-4
14. Mao Z, Li H, Du B, Cui K, Xing Y, Zhao X, Zai S (2017) LncRNA dancr promotes migration and invasion through suppression of lncRNA-let in gastric cancer cells. Biosci Rep. https://doi.org/10.1042/BSR20171070
15. Zhao Q, Yu H, Ming Z, Hu H, Ren G, Liu H (2018) The bipartite network projection-recommended algorithm for predicting long non-coding RNA-protein interactions. Mol Therapy Nucleic Acids 13:464–471. https://doi.org/10.1016/j.omtn.2018.09.020
16. Ge M, Li A, Wang M (2016) A bipartite network-based method for prediction of long non-coding RNA-protein interactions. Genom Proteom Bioinform 14(1):62–71. https://doi.org/10.1016/j.gpb.2016.01.004
17. Zhou YK, Hu J, Shen ZA, Zhang WY, Du PF (2020) Lpi-skf: Predicting lncRNA-protein interactions using similarity kernel

fusions. Front Genet 11:1554. https://doi.org/10.3389/fgene.2020.615144
18. Zheng X, Wang Y, Tian K, Zhou J, Guan J, Luo L, Zhou S (2017) Fusing multiple protein-protein similarity networks to effectively predict lncRNA-protein interactions. BMC Bioinform 18(12):11–18. https://doi.org/10.1186/s12859-017-1819-1
19. Liu H, Ren G, Hu H, Zhang L, Ai H, Zhang W, Zhao Q (2017) Lpi-nrlmf: lncrna-protein interaction prediction by neighborhood regularized logistic matrix factorization. Oncotarget. https://doi.org/10.18632/oncotarget.21934
20. Zhang T, Wang M, Xi J, Li A (2018) Lpgnmf: predicting long non-coding RNA and protein interaction using graph regularized nonnegative matrix factorization. IEEE/ACM Trans Comput Biol Bioinform 17(1):189–197. https://doi.org/10.1109/TCBB.2018.2861009
21. Ma Y, He T, Jiang X (2019) Projection-based neighborhood nonnegative matrix factorization for lncRNA-protein interaction prediction. Front Genet 10:1148. https://doi.org/10.3389/fgene.2019.01148
22. Hu H, Zhang L, Ai H, Zhang H, Fan Y, Zhao Q, Liu H (2018) HLPI-ensemble: prediction of human LNCRNA-protein interactions based on ensemble strategy. RNA Biol 15(6):797–806. https://doi.org/10.1080/15476286.2018.1457935
23. Zhang W, Yue X, Tang G, Wu W, Huang F, Zhang X (2018) Sfpel-lpi: sequence-based feature projection ensemble learning for predicting lncrna-protein interactions. PLoS Comput Biol 14(12):e1006616. https://doi.org/10.1371/journal.pcbi.1006616
24. Deng L, Wang J, Xiao Y, Wang Z, Liu H (2018) Accurate prediction of protein-LNCRNA interactions by diffusion and hetesim features across heterogeneous network. BMC Bioinform 19(1):1–11. https://doi.org/10.1186/s12859-018-2390-0
25. Fan XN, Zhang SW (2019) LPI-BLS: predicting LNCRNA-protein interactions with a broad learning system-based stacked ensemble classifier. Neurocomputing 370:88–93. https://doi.org/10.1016/j.neucom.2019.08.084
26. Wekesa JS, Meng J, Luan Y (2020) Multi-feature fusion for deep learning to predict plant lncRNA-protein interaction. Genomics 112(5):2928–2936. https://doi.org/10.1016/j.ygeno.2020.05.005
27. Yi HC, You ZH, Wang MN, Guo ZH, Wang YB, Zhou JR (2020) Rpi-se: a stacking ensemble learning framework for ncrna-protein interactions prediction using sequence information. BMC Bioinform 21(1):1–10. https://doi.org/10.1186/s12859-020-3406-0
28. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.4258/hir.2016.22.4.351
29. Chu Y, Kaushik AC, Wang X, Wang W, Zhang Y, Shan X, Salahub DR, Xiong Y, Wei DQ (2019) DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features. Brief Bioinform. https://doi.org/10.1093/bib/bbz152
30. Kaushik AC, Wang YJ, Wang X, Kumar A, Singh SP, Pan CT, Shiue YL, Wei DQ (2019) Evaluation of anti-EGFR-IRGD recombinant protein with gold nanoparticles: synergistic effect on antitumor efficiency using optimized deep neural networks. RSC Adv 9(34):19261–19270. https://doi.org/10.1039/C9RA01975H
31. Gainza P, Sverrisson F, Monti F, Rodola E, Boscaini D, Bronstein M, Correia B (2020) Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nat Methods 17(2):184–192. https://doi.org/10.1038/s41592-019-0666-6
32. Deng L, Wu H, Liu H (2019) D2vcb: A hybrid deep neural network for the prediction of in-vivo protein-DNA binding from combined DNA sequence. In: 2019 IEEE International Conference on bioinformatics and biomedicine (BIBM). IEEE, pp 74–77. https://doi.org/10.1109/BIBM47256.2019.8983051
33. Zhang Y, Jia C, Kwoh CK (2020) Predicting the interaction biomolecule types for lncRNA: an ensemble deep learning approach. Brief Bioinform. https://doi.org/10.1093/bib/bbaa228
34. Wekesa JS, Meng J, Luan Y (2020) A deep learning model for plant lncRNA-protein interaction prediction with graph attention. Mol Genet Genom 295(5):1091–1102. https://doi.org/10.1007/s00438-020-01682-w
35. Wei H, Liao Q, Liu B (2020) ilncrnadis-fb: identify lncRNA-disease associations by fusing biological feature blocks through deep neural network. IEEE/ACM Trans Comput Biol Bioinform. https://doi.org/10.1109/TCBB.2020.2964221
36. Zhao T, Hu Y, Peng J, Cheng L (2020) Deeplgp: a novel deep learning method for prioritizing lncrna target genes. Bioinformatics 36(16):4466–4472. https://doi.org/10.1093/bioinformatics/btaa428
37. Shaw D, Chen H, Xie M, Jiang T (2021) Deeplpi: a multimodal deep learning method for predicting the interactions between lncrnas and protein isoforms. BMC Bioinform 22(1):1–22. https://doi.org/10.1186/s12859-020-03914-7
38. Li A, Ge M, Zhang Y, Peng C, Wang M (2015) Predicting long noncoding RNA and protein interactions using heterogeneous network model. BioMed Res Int. https://doi.org/10.1155/2015/671950
39. Yuan J, Wu W, Xie C, Zhao G, Zhao Y, Chen R (2014) Npinter v2. 0: an updated database of ncrna interactions. Nucleic Acids Res 42(D1):D104–D108. https://doi.org/10.1093/nar/gkt1057
40. Xie C, Yuan J, Li H, Li M, Zhao G, Bu D, Zhu W, Wu W, Chen R, Zhao Y (2014) Noncodev4: exploring the world of long noncoding RNA genes. Nucleic Acids Res 42(D1):D98–D103. https://doi.org/10.1093/nar/gkt1222
41. Consortium U (2019) Uniprot: a worldwide hub of protein knowledge. Nucleic Acids Res 47(D1):D506–D515. https://doi.org/10.1093/nar/gky1049
42. Zhang W, Qu Q, Zhang Y, Wang W (2018) The linear neighborhood propagation method for predicting long non-coding RNA-protein interactions. Neurocomputing 273:526–534. https://doi.org/10.1016/j.neucom.2017.07.065
43. Bai Y, Dai X, Ye T, Zhang P, Yan X, Gong X, Liang S, Chen M (2019) PLNCRNADB: a repository of plant LNCRNAS and LNCRNA-RBP protein interactions. Curr Bioinform 14(7):621–627. https://doi.org/10.2174/1574893614666190131161002
44. Shrikumar A, Prakash E, Kundaje A (2019) Gkmexplain: fast and accurate interpretation of nonlinear gapped k-mer svms. Bioinformatics 35(14):i173–i182. https://doi.org/10.1093/bioinformatics/btz322
45. Tahir M, Hayat M, Khan SA (2019) inuc-ext-psetnc: an efficient ensemble model for identification of nucleosome positioning by extending the concept of chou's pseaac to pseudo-tri-nucleotide composition. Mol Genet Genom 294(1):199–210. https://doi.org/10.1007/s00438-018-1498-2
46. Liu B, Liu F, Fang L, Wang X, Chou KC (2015) REPDNA: a python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. Bioinformatics 31(8):1307–1309. https://doi.org/10.1093/bioinformatics/btu820
47. Su Y, Luo Y, Zhao X, Liu Y, Peng J (2019) Integrating thermodynamic and sequence contexts improves protein-RNA binding prediction. PLoS Comput Biol 15(9):e1007283. https://doi.org/10.1371/journal.pcbi.1007283
48. Magnan CN, Baldi P (2014) Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics 30(18):2592–2597. https://doi.org/10.1093/bioinformatics/btu352
49. Peterson LE (2009) K-nearest neighbor. Scholarpedia 4(2):1883. https://doi.org/10.4249/scholarpedia.1883

50. Du Q, Faber V, Gunzburger M (1999) Centroidal voronoi tessellations: applications and algorithms. SIAM Rev 41(4):637–676. https://doi.org/10.1137/S0036144599352836
51. Balsubramani A, Dasgupta S, Freund Y, Moran S (2019) An adaptive nearest neighbor rule for classification. In: NeurIPS, pp 7577–7586. https://par.nsf.gov/biblio/10168808
52. Zhao T, Hu Y, Valsdottir LR, Zang T, Peng J (2020) Identifying drug-target interactions based on graph convolutional network and deep neural network. Brief Bioinform. https://doi.org/10.1093/bib/bbaa044
53. Wang L, You ZH, Huang YA, Huang DS, Chan KC (2020) An efficient approach based on multi-sources information to predict circrna-disease associations using deep convolutional neural network. Bioinformatics 36(13):4038–4046. https://doi.org/10.1093/bioinformatics/btz825
54. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
55. Cai L, Lu C, Xu J, Meng Y, Wang P, Fu X, Zeng X, Su Y (2021) Drug repositioning based on the heterogeneous information fusion graph convolutional network. Brief Bioinform. https://doi.org/10.1093/bib/bbab319
56. Zhou ZH, Feng J (2019) Deep forest. National Sci Rev 6(1):74–86. https://doi.org/10.1093/nsr/nwy108
57. Zhou ZH, Feng J (2017) Deep forest[J]. arXiv preprint arXiv:1702.08835
58. Qi Y (2012) Random forest for bioinformatics. In: Ensemble machine learning. Springer, New York, pp 307–323. https://doi.org/10.1007/978-1-4419-9326-7_11
59. Biau G, Scornet E (2016) A random forest guided tour. Test 25(2):197–227. https://doi.org/10.1007/s11749-016-0481-7
60. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63(1):3–42. https://doi.org/10.1007/s10994-006-6226-1
61. Cao Y, Geddes TA, Yang JYH, Yang P (2020) Ensemble deep learning in bioinformatics. Nat Mach Intell 2(9):500–508. https://doi.org/10.1038/s42256-020-0217-y
62. Chen X, Zhu CC, Yin J (2019) Ensemble of decision tree reveals potential mirna-disease associations. PLoS Comput Biol 15(7):e1007209. https://doi.org/10.1371/journal.pcbi.1007209
63. Chen X, Xie D, Zhao Q, You ZH (2019) Micrornas and complex diseases: from experimental results to computational models. Brief Bioinform 20(2):515–539. https://doi.org/10.1093/bib/bbx130
64. Hu Y, Ma Z, He Y, Liu W, Su Y, Tang Z (2017) LNCRNA-SNHG1 contributes to gastric cancer cell proliferation by regulating DNMT1. Biochem Biophys Res Commun 491(4):926–931. https://doi.org/10.1016/j.bbrc.2017.07.137
65. Zhang M, Wang W, Li T, Yu X, Zhu Y, Ding F, Li D, Yang T (2016) Long noncoding RNA snhg1 predicts a poor prognosis and promotes hepatocellular carcinoma tumorigenesis. Biomed Pharmacother 80:73–79. https://doi.org/10.1016/j.biopha.2016.02.036
66. Cao B, Wang T, Qu Q, Kang T, Yang Q (2018) Long noncoding rna snhg1 promotes neuroinflammation in parkinson's disease via regulating mir-7/nlrp3 pathway. Neuroscience 388:118–127. https://doi.org/10.1016/j.neuroscience.2018.07.019
67. Cui Y, Zhang F, Zhu C, Geng L, Tian T, Liu H (2017) Upregulated LNCRNA SNHG1 contributes to progression of non-small cell lung cancer through inhibition of MIR-101-3p and activation of wnt/ β-catenin signaling pathway. Oncotarget 8(11):17785. https://doi.org/10.18632/oncotarget.14854
68. Lan T, Yuan K, Yan X, Xu L, Liao H, Hao X, Wang J, Liu H, Chen X, Xie K et al (2019) LNCRNA SNHG10 facilitates hepatocarcinogenesis and metastasis by modulating its homolog scarna13 via a positive feedback loop. Can Res 79(13):3220–3234. https://doi.org/10.1158/0008-5472
69. Yuan X, Yang T, Xu Y, Ou S, Shi P, Cao M, Zuo X, Liu Q, Yao J (2020) Snhg10 promotes cell proliferation and migration in gastric cancer by targeting mir-495-3p/ctnnb1 axis. Dig Dis Sci:1–10. https://doi.org/10.1007/s10620-020-06576-w
70. Liang M, Wang L, Cao C, Song S, Wu F (2020) LNCRNA SNHG10 is downregulated in non-small cell lung cancer and predicts poor survival. BMC Pulm Med 20(1):1–6. https://doi.org/10.1186/s12890-020-01281-w
71. Zhu S, Liu Y, Wang X, Wang J, Xi G (2020) Lncrna snhg10 promotes the proliferation and invasion of osteosarcoma via wnt/ β-catenin signaling. Mol Therapy Nucleic Acids 22:957–970. https://doi.org/10.1016/j.omtn.2020.10.010
72. Li J, Sun W (2018) Exploration of radiosensitivity-related LNCRNAS in esophageal cancer stem cell. Int J Radiat Oncol Biol Phys 102(3):e33. https://doi.org/10.1016/j.ijrobp.2018.07.524
73. Chen X, Wang L, Qu J, Guan NN, Li JQ (2018) Predicting mirna-disease association based on inductive matrix completion. Bioinformatics 34(24):4256–4265. https://doi.org/10.1093/bioinformatics/bty503