frontiers | Frontiers in Microbiology

TYPE Original Research
PUBLISHED 05 January 2023
DOI 10.3389/fmicb.2022.1092467

OPEN ACCESS

EDITED BY
Lihong Peng,
Hunan University of Technology, China

REVIEWED BY
Yuhua Yao,
Hainan Normal University,
China
Yi Xiong,
Shanghai Jiao Tong University, China

*CORRESPONDENCE
Zhixiang Yin
✉ zxyin66@163.com

SPECIALTY SECTION
This article was submitted to Systems Microbiology, a section of the journal Frontiers in Microbiology

RECEIVED 08 November 2022
ACCEPTED 07 December 2022
PUBLISHED 05 January 2023

CITATION
Peng Y, Zhao S, Zeng Z, Hu X and Yin Z (2023) LGBMDF: A cascade forest framework with LightGBM for predicting drug-target interactions.
Front. Microbiol. 13:1092467.
doi: 10.3389/fmicb.2022.1092467

LGBMDF: A cascade forest framework with LightGBM for predicting drug-target interactions

Yu Peng, Shouwei Zhao, Zhiliang Zeng, Xiang Hu and Zhixiang Yin*

School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, Shanghai, China

Prediction of drug-target interactions (DTIs) plays an important role in drug development. However, traditional laboratory methods to determine DTIs require a lot of time and capital costs. In recent years, many studies have shown that using machine learning methods to predict DTIs can speed up the drug development process and reduce capital costs. An excellent DTI prediction method should have both high prediction accuracy and low computational cost. In this study, we noticed that the previous research based on deep forests used XGBoost as the estimator in the cascade, we applied LightGBM instead of XGBoost to the cascade forest as the estimator, then the estimator group was determined experimentally as three LightGBMs and three ExtraTrees, this new model is called LGBMDF. We conducted 5-fold cross-validation on LGBMDF and other state-of-the-art methods using the same dataset, and compared their Sn, Sp, MCC, AUC and AUPR. Finally, we found that our method has better performance and faster calculation speed.

KEYWORDS

drug-target interactions, machine learning, LightGBM, deep forest, prediction

1. Introduction

In recent years, with the rapid development of computer data processing capabilities, the continuous enrichment of data content, and the improvement of algorithm models, more and more researches on artificial intelligence in the fields of biology and medicine have been carried out (Guo et al., 2021; Chen and Yin, 2022; Zhou et al., 2022). Many computational methods based on machine learning have been proposed to solve biological problems (Lihong et al., 2021; Zhou et al., 2021; Peng et al., 2022; Shen et al., 2022). Especially in drug development, the prediction of drug-target interactions (DTIs) played an important role in drug development and drug repositioning, so using machine learning methods to predict DTIs became a research hotspot.

Over the past decade, a large number of machine learning-based methods were proposed for identifying DTI (Zhou et al., 2019). Among them, binary classification

methods account for the majority. Some methods identify drug-target pairs based on drug and protein information, Li et al. (2020) used protein sequences and drug substructure fingerprint information to predict DTIs. In addition, there were many models (Mousavian et al., 2016; Li et al., 2020; Zhan et al., 2020; Tanoori et al., 2021) that predicted new DTIs based on information similarity.

In fact, there are more methods based on network inference, Yamanishi et al. (2010) integrated chemical, genomic and pharmacological information in bipartite graph to uncover potential DTIs. Mei J. P et al. (Mei et al., 2013) proposed Neighbor-based Interaction-profile Inferring (NII) based on bipartite local model (BLM). Chen et al. (2012) proposed the method of Network-based Random Walk with Restart on the Heterogeneous network (NRWRH) which integrates three different networks into a heterogeneous network through known DTIs, and achieves random wandering on this heterogeneous network. Cao et al. (2014) proposed a computational method for DTI prediction by combining the information from chemical, biological, and network properties. Ding et al. (2017) used molecular substructure fingerprints, multivariate mutual information (MMI) of proteins and network topology to represent drugs, targets and their relationships, and employ SVM and Feature Selection (FS) to build predictive models. Thereafter, scholars began to extract features from more complex networks. SNF-CVAE (Jarada et al., 2021) integrates similarity network fusion (SNF) and collective variational autoencoder (CVAE) to improve prediction accuracy. An and Yu (2021) proposed a Network Embedding framework in mulTiPlex networks (NEDTP) to predict DTIs. Jin et al. (2021) proposed a machine learning model called HeTDR, the method combines drug features in multiple networks and disease features in biomedical corpora to predict the degree of association between drugs and diseases. In addition, there are some computational methods based on matrix factorization (Gönen, 2012; Liu et al., 2016; Bagherian et al., 2021) and multi-label learning (Yuan et al., 2016; Pliakos et al., 2019; Chu et al., 2021b).

Moreover, with the rise of deep learning methods, people have made a lot of achievements in the field of DTI prediction based on deep learning methods. Many scholars consider graph analysis (Olayan et al., 2018; Peng et al., 2021; Yang et al., 2022) as an important means to predict DTIs. Many models apply deep neural networks (DNN) to DTI prediction, LASSO-DNN (You et al., 2019) combines LASSO with DNN, deepDTnet (Zeng et al., 2020b) applies DNN algorithm to network embedding, DeepFusionDTA (Pu et al., 2021) proposes a two-stage deep neural network ensemble model, based on DNN, DNN-DTIs (Chen et al., 2021) employs layer-by-layer learning method to predict DTIs. Besides, DeepACTION (Hasan Mahmud et al., 2020), AutoDTI++ (Sajadi et al., 2021), GCNMK (Wang et al., 2022) and DeepStack-DTIs (Zhang et al., 2022) also use deep learning methods.

Specially, inspired by DNN, Zhou and Feng (2017) proposed Deep Forest, and some DTI prediction methods based on Deep Forest showed good performance. Such as AOPEDF (Zeng et al., 2020a), DTI-CDF (Chu et al., 2021a) and EC-DFR (Lin et al., 2022).

In this study, we make some improvements based on the AOPEDF model, thus proposing a new method termed LGBMDF. We add LightGBM (Ke et al., 2017), which outperforms XGBoost and CatBoost in another work (Al Daoud, 2019), to Cascade Forest as a new estimator. For the convenience of comparison, we used the same feature extraction method as AOPEDF. For the obtained vector features, we input them into a modified Cascade Forest for predicting DTIs. Finally, we compared our model with other models in terms of performance and speed, our model is comparable to and in some way ahead of the state-of-the-art models. In conclusion, LGBMDF is a very practical method for DTI prediction, which can help new drug development and some other fields, such as identifying miRNA-disease associations or the associations between cancers and microbes.

2. Materials and methods

2.1. Data resource

DTI-related information was collected from DrugBank (v4.2) (Wishart et al., 2018), the Therapeutic Target Database (Yang et al., 2016), and the PharmGKB (Hernandez-Boussard et al., 2007) database. Bioactivity data for drug–target pairs are collected from ChEMBL (v20) (Gaulton et al., 2012), BindingDB (Liu et al., 2007), and IUPHAR/BPS Guide to PHARMACOLOGY (Pawson et al., 2014). The chemical structure of each drug with SMILES format is extracted from DrugBank (v4.0) (Law et al., 2014). Here, only DTIs meeting the following three criteria are used: (i) the human target is represented by a unique UniProt (Apweiler et al., 2004) accession number; (ii) the target is marked as 'reviewed' in the UniProt database; (iii) binding affinities, all the $K_i, K_d, IC50\ or\ EC50 \leq 10\mu M$. In short, we constructed a DTI network by using 732 FDA-approved drugs and 1915 targets. In addition, we used 9 drug-related networks and 6 protein-related networks (Cheng et al., 2019a,b; Zeng et al., 2020a). For the feature extraction approach, in order to facilitate comparison, we referred to the previous studies (Zhang et al., 2018; Zeng et al., 2020a).

2.2. Deep forest

The deep neural network has shown good performance in many works. Inspired by DNN, Zhou and Feng (2017) proposed an ensemble algorithm with deep structure based on decision tree. It has much fewer hyperparameters than DNNs, and the complexity of the model can be automatically determined based on the input variables.

After obtaining low-dimensional vector representations of drugs and proteins (targets), we input them into Cascade Forest to predict DTIs. In the cascade structure, the output features vector of the previous layer and the original features vector is used as the

input features vector of the next layer. Furthermore, when a new layer is generated, the performance of the entire cascade is estimated on the validation set, and the training process is terminated if there is no significant increase in performance. The estimators setting at each layer are also important, after experimental testing, we set up three ExtraTrees and three LightGBMs (Figure 1).

To prevent overfitting, class vectors for each estimator are generated by *k*-fold cross-validation. Specifically, the average of the generated *k*-1 class vectors is obtained to obtain the final class vector as the enhanced feature of the next layer.

2.3. LightGBM classifier

2.3.1. Histogram algorithm

The basic idea: First, the continuous floating-point feature values are discretized into k integers, and a histogram of width k is constructed (Figure 2). When the samples are traversed once, the histogram accumulates the required statistics and then traverses the histogram to find the optimal partition point based on the discrete values of the histogram.

Another improved speedup of LightGBM is to subtract the histogram of sibling nodes from the histogram of the parent node so that the speed can be doubled (Figure 3). Usually, when constructing a histogram, it is necessary to traverse all the data on that leaf, but histogram differencing only requires traversing k bins of the histogram. In the actual process of constructing the tree, LightGBM can also calculate the smaller leaf nodes of the histogram first, and then use histogram difference to obtain the larger leaf nodes of the histogram, so that we can get the histogram of its sibling leaf at a very small cost.

2.3.2. Leaf-wise algorithm with depth restriction

Based on the histogram algorithm, LightGBM is further optimized. First, it abandons the level-wise (Figure 4A) tree growth strategy used by most GBDT algorithms and applies the leaf-wise tree growth (Figure 4B) with depth restriction. XGBoost uses level-wise growth strategy, which can split the leaves of the same level at the same time by traversing the data once, making it easy to perform multi-threaded optimization and control the model complexity without overfitting. However, level-wise is an inefficient algorithm because it treats the leaves of the same layer indiscriminately, and in fact, many leaves have low splitting gain, so there is no need to split, thus bringing a lot of unnecessary computational overhead. LightGBM uses

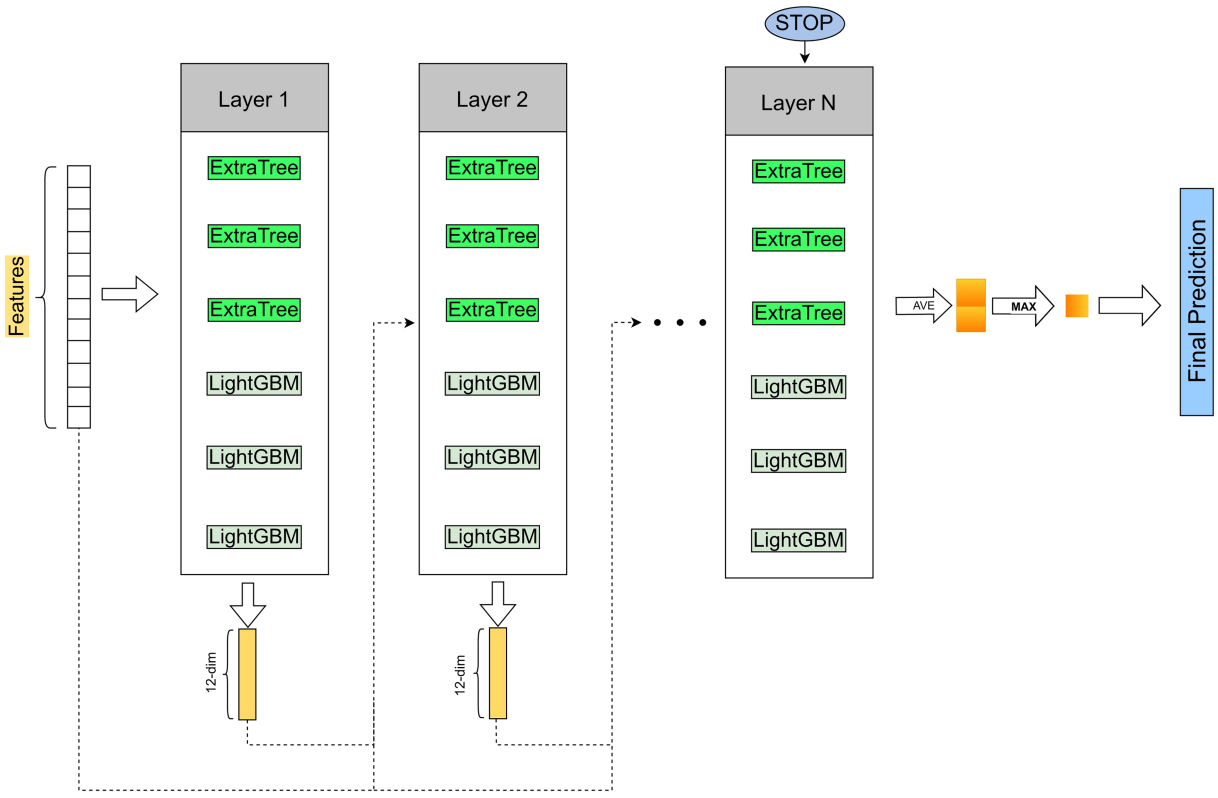

FIGURE 1
The pipeline of LGBMDF. After getting the features of drugs and targets, we process these features with cascade forest, and set 3 LightGBMs and 3 ExtraTrees for each level as estimators, each estimator outputs a 2-dimensional class vector, and then concatenate the output class vector and the original feature vector as the input vector for the next layer.

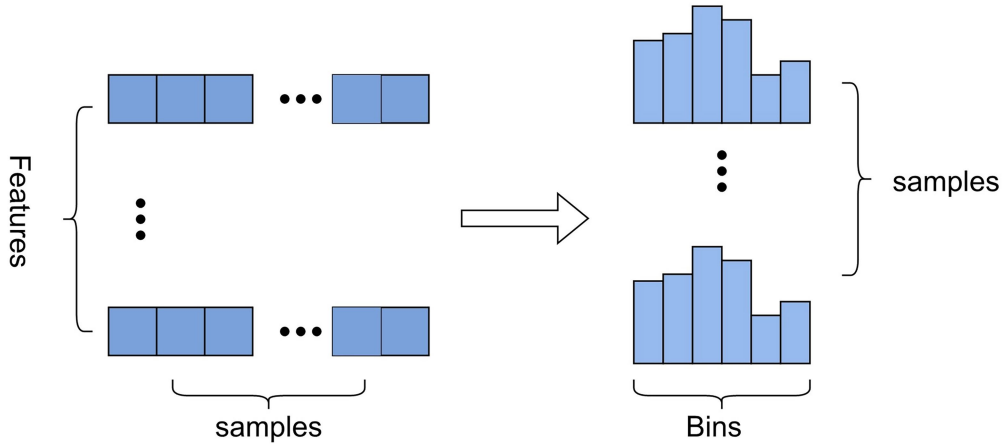

FIGURE 2
The construction of histogram.

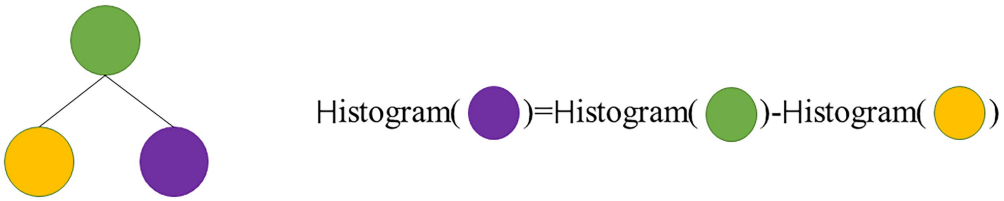

FIGURE 3
Subtract the histogram of sibling node from the histogram of the parent node so that the speed can be doubled.

leaf-wise tree growth strategy, which can locate the leaf with the largest splitting gain from all the current leaves, and then splits it, cycling as this way. Therefore, compared with level-wise, the advantage of leaf-wise is that it can reduce more errors and get better accuracy with the same number of splits; the disadvantage of leaf-wise is that it may grow a deeper decision tree and produce overfitting. For this reason, LightGBM adds a maximum depth limit to leaf-wise to ensure high efficiency and prevent overfitting at the same time.

2.3.3. Gradient-based one-side sampling

The feature vector in Adaboost can represent the importance of a sample well, but there is not a weight vector like this one in GBDT. Fortunately, we found that the sample gradient of GBDT is a good indicator, and samples with small gradients will have small training errors and have been well-trained. Generally, the simpler idea is to discard samples with small gradients, but this will affect the model performance, thus we propose a new method named gradient-based one-side sampling (GOSS).

The basic idea of GOSS is to reduce the complexity of the model by reducing the sample size. GOSS first sorts the samples by the gradient from largest to smallest, uses the top-ranked $a\times100\%$, and then randomly samples the rest data with small gradients $b\times100\%$. Then GOSS amplifiers the data with a small gradient by a constant $\frac{1-a}{b}$ when calculating the information gain.

In GBDT, we assume the input space as X^s, the gradient space as G. Suppose that there are n i.i.d instances $\{x_1,x_2,\cdots,x_n\}$, x_i is a vector of dimension s in X^s. The negative gradient of the loss function is represented as $\{g_1,g_2,\cdots,g_n\}$. The Decision tree model splits nodes where information gain is the largest, and the information gain is usually determined by the variance after the split.

Let $\mathcal{O}$ be the training set of a node d on the decision tree, and the variance of the split feature j at this point is defined as:

$$V_{j|\mathcal{O}}(d)=\frac{1}{n_{\mathcal{O}}}\left(\frac{\left(\sum_{\{x_i\in\mathcal{O}:x_{ij}\le d\}}g_i\right)^2}{n^j_{l|\mathcal{O}}(d)}+\frac{\left(\sum_{\{x_i\in\mathcal{O}:x_{ij}>d\}}g_i\right)^2}{n^j_{r|\mathcal{O}}(d)}\right) \quad (1)$$

Where $n_{\mathcal{O}}=\sum I[x_i\in\mathcal{O}], n^j_{l|\mathcal{O}}(d)=\sum I[x_i\in\mathcal{O}:x_{ij}\le d]$ and $n^j_{r|\mathcal{O}}(d)=\sum I[x_i\in\mathcal{O}:x_{ij}>d]$.

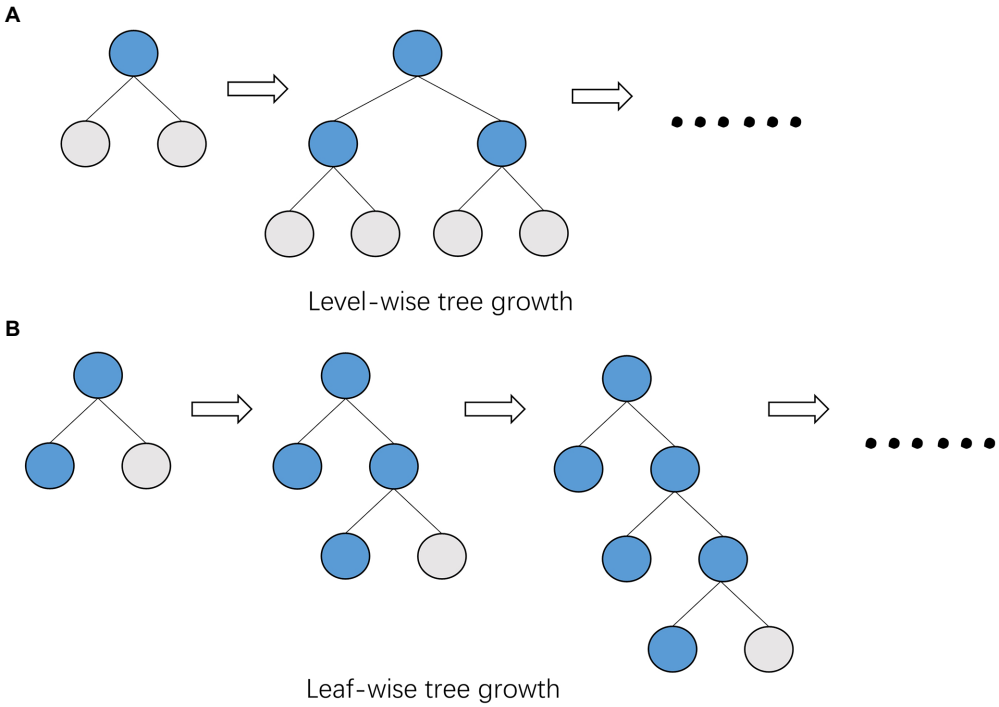

FIGURE 4
Comparison of tree growth patterns between XGBoost and LightGBM. **(A)** XGBoost uses the level-wise growth strategy, which can split the leaves of the same level at the same time by traversing the data once. **(B)** LightGBM uses the leaf-wise growth strategy, which finds the leaf with the largest splitting gain from all the current leaves, and then splits it.

In GOSS, First, all instances absolute values of gradients are sorted in descending order. We select the first $a \times 100\%$ samples as set A, and then randomly sample B of size $b \times \#A^c$ | from the remaining instance set A^c. Finally, we split the instance *via* estimated variance $\widetilde{V}_j(d)$ on $A \cup B$.

$$\widetilde{V}_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b}\sum_{x_i \in B_l} g_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b}\sum_{x_i \in B_r} g_i\right)^2}{n_r^j(d)}\right) \quad (2)$$

Where $A_l = \{x_i \in A : x_{ij} \le d\}, A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \le d\}, B_r = \{x_i \in B : x_{ij} > d\}$. $\frac{1-a}{b}$ is to normalize the size of B to the size of A^c.

2.3.4. Exclusive feature bundling

High-dimensional space is always sparse, and in a sparse feature space, many features are mutually exclusive, so we can bind mutually exclusive features into a single feature (Figure 5). Through the feature scanning algorithm, we can use the designed feature scanning algorithm to construct the same histogram from the feature bundles as the original single feature. In this way, we can decrease the complexity of histogram building from $O(\#sample \times \#feature)$ to $O(\#sample \times \#bundle)$, while $\#bundle \ll \#feature$, thus we can greatly improve the training speed of GBDT.

In general, compare to XGBoost, LightGBM has the advantages of faster speed and smaller memory usage. LightGBM uses the histogram algorithm to transform the traversal samples into traversal histograms, which greatly reduces the time complexity; applies the GOSS algorithm to filter out many samples with small gradients and adopts leaf-wise growth strategy to build the trees, which reduces a lot of unnecessary calculations. In addition, LightGBM utilizes EFB algorithm to decrease the number of features.

2.4. Evaluation metric

To compare with other methods, we perform a 5-fold cross-validation and adopt Sn, Sp, MCC, AUC and AUPR as evaluation metrics.

Sn, Sp and MCC are commonly used evaluation indicators for binary classification problems, and their calculations are based on the confusion matrix.

$$S_n = \frac{TP}{TP + FN} \quad (3)$$

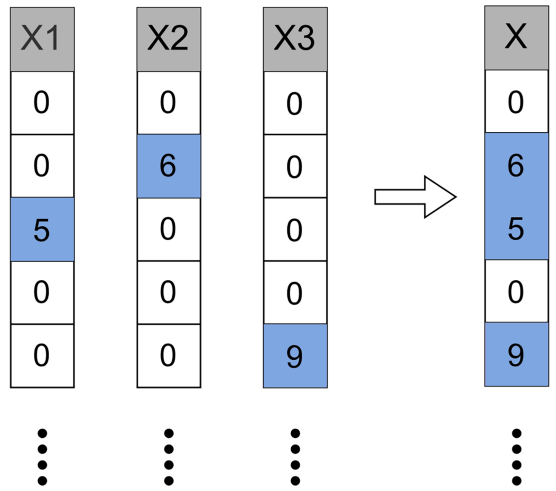

FIGURE 5
Bind mutually exclusive features into a single feature.

$$S_p = \frac{TN}{TN + FP} \tag{4}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

Receiver operating characteristic (ROC) curve is often used to evaluate the model's prediction performance. It is calculated based on the confusion matrix. The higher the curve on the upper left, the better the performance of the model. The vertical axis of the ROC curve is the "True Positive Rate," and the horizontal axis is the "False Positive Rate," which are, respectively, defined as:

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

$$FPR = \frac{FP}{TN + FP} \tag{7}$$

However, the ROC curves of some models will cross, so we generally choose the AUC (Area Under ROC Curve) for comparison. We assume that the points of the ROC curve are connected in order by the points of $\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$, then the AUC can be estimated as:

$$AUC = \frac{1}{2}\sum_{i=0}^{m-1}(x_{i+1} - x_i) \bullet (y_{i+1} - y_i) \tag{8}$$

The PR curve represents the relationship between Precision and Recall. In general, Recall is set to the abscissa and Precision is set to the ordinate. Precision and Recall can be calculated according to the confusion matrix.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

AUPR is the Area Under PR curve. In such a highly imbalanced dataset, AUPR can provide better performance evaluation because it penalizes false positives more severely.

3. Results

3.1. Parameter optimization

We optimized the parameters of the estimators, considering the impact of parameters on model performance. By the means of employing GridSearchCV function, we set the interval of the parameter, the "scoring" is set as "accuracy." The parameter optimization results are shown in Table 1.

3.2. Estimators setting for each layer

When reproducing the AOPEDF model, we noticed that the XGBoost in cascade is time-consuming, so we chose LightGBM, a classifier that performs better than XGBoost in another work (Al Daoud, 2019), as estimator to accelerate the calculation speed of the model and reduce the computing cost and time cost. We tested five combinations and compared their Sn, Sp, MCC, AUC, AUPR (Table 2) and running time. The experiments are run in the environment of Python3.9, CPU: 2* Intel (R) Xeon (R) Gold 6320R, RAM: 128G.

The names of each combination in the Figure 6 are explained as follows:

- *AOPEDF*: 2 ExtraTrees, 2 RFs and 2 XGBoosts
- *2LGB-2RF-2ET*: 2 LightGBMs, 2 RFs and 2 ExtraTrees
- *3LGB-3RF*: 3 LightGBMs and 3 RFs
- *3LGB-3ET*: 3 LightGBMs and 3 ExtraTrees.

After experiments, we found that the MCC, AUC and AUPR values of 3LGB-3ET are higher than that of the others. Moreover, the calculation speed of 3LGB-3ET is more than twice as fast as AOPEDF. Therefore, we choose the combination of 3LGB-3ET to set the estimators for each layer finally.

3.3. Model comparison

The following 4 models were adopted as baseline methods.

NEDTP (An and Yu, 2021): A node similarity network is constructed based on 15 heterogeneous information networks, and then random walks are applied to extract the topology information of each node in the network and learn it as a low-dimensional vector. Finally, employ LightGBM algorithm to complete the classification task.

AOPEDF (Zeng et al., 2020a): It integrates 15 biological networks to construct a heterogeneous network, and then learns low-dimensional vector representations of features from this heterogeneous network that keep arbitrary-order proximity. Then use the deep forest to predict new DTIs.

Random Forest (Breiman, 2001): It is a combination of tree predictors such that each tree depends on the value of an independently sampled random vector and all trees in the forest have the same distribution.

Support Vector Machine, SVM (Vapnik and Chervoneva, 1964): It is a class of generalized linear classifiers for binary classification of data in a supervised learning manner.

TABLE 1 The result of parameter optimization.

Model	Parameter	Range	Used
RandomForest	n_estimators	[100, 200, 400, 500, 600]	400
LightGBM	n_estimators	[100, 200, 400, 500]	400
	max_depth	[7, 8, 9, 10, 11]	11
	num_leaves	[100, 200, 300, 400, 500]	200
ExtraTree	n_estimators	[100, 200, 400, 500, 600]	500

TABLE 2 Performance comparison under each estimator setting.

Estimators	Sn	Sp	MCC	AUC	AUPR
AOPEDF	**0.9463**	0.9447	0.8911	0.9842	0.9855
2LGB-2RF-2ET	0.9439	**0.9477**	0.8918	0.9841	0.9854
3LGB-3RF	0.9443	0.9453	0.8898	0.9839	0.9849
3LGB-3ET	0.9451	0.9471	**0.8924**	**0.9844**	**0.9857**

The bold values represent the maximum value of each estimator setting under each evaluation metric.

We took drug-protein pairs with known interactions as positive samples, and pairs with unknown interactions as negative samples, and then selected all positive samples and randomly sampled negative samples with the same number of positive samples for 5-fold cross-validation to evaluate model performance (Figure 7, Table 3). For each 5-fold cross-validation, we select 80% positive pairs and the corresponding number of randomly sampled negative pairs as the training set, and the remaining 20% positive pairs and the corresponding number of randomly sampled negative pairs as the test set. We found that the Sp, MCC, AUC, and AUPR of LGBMDF are all higher than those of other methods. In addition, in previous experiments, we have found that LGBMDF is faster than AOPEDF. An excellent model needs to consider both the accuracy and the computing power cost of the model. Therefore, our model is better than the current advanced model in general.

4. Discussion

This paper investigated the application of machine learning methods for DTI prediction. Traditional drug-target effect testing methods are time-consuming and labor-intensive. And Machine learning methods have attracted the attention of many researchers due to these methods can greatly reduce the related costs. We chose the same feature extraction method as AOPEDF, and used this method to extract low-dimensional representations of drug and protein features from 15 biological networks, and these features maintain arbitrary order proximity.

After obtaining low-dimensional feature representations of drugs and targets, we used cascaded deep forests for DTI prediction. Specifically, we used LightGBM as the estimator in the cascade to reduce the computational cost. And the LightGBM has shown better performance and computational speed than XGBoost in other experiments. Considering the effect of estimator diversity in the cascade, we also chose ExtraTree as the estimator.

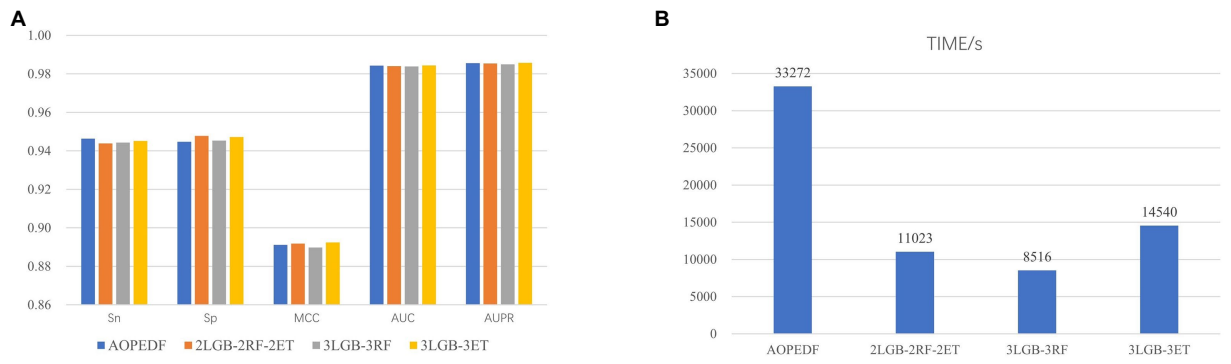

FIGURE 6
Model performance comparison under each estimator setting. **(A)** AUC and AUPR for 4 estimator combinations. **(B)** Computational time for 4 estimator combinations.

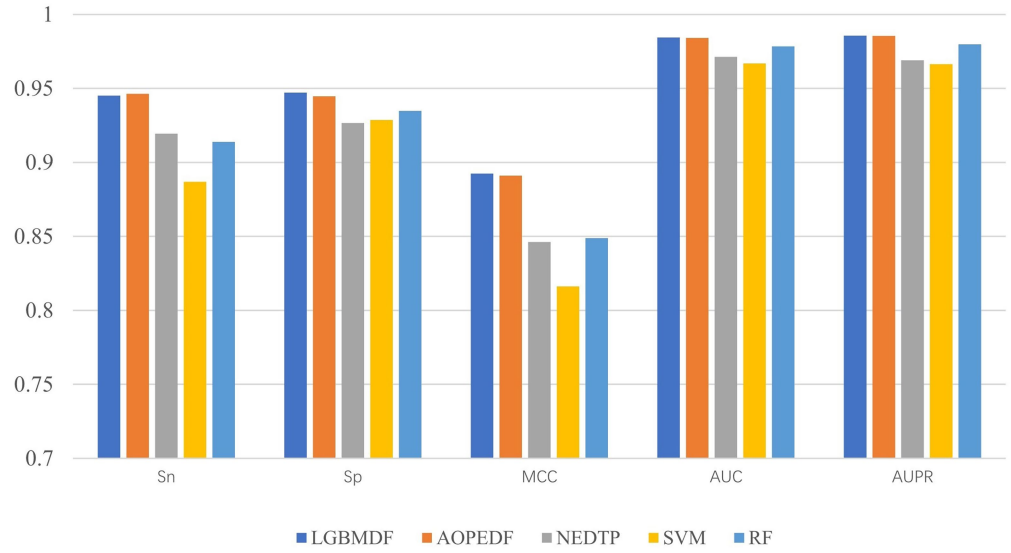

FIGURE 7
Sn, Sp, MCC, AUC and AUPR of LGBMDF, AOPEDF, NEDTP, RF, SVM.

TABLE 3 Performance of LGBMDF and baseline methods.

Model	Sn	Sp	MCC	AUC	AUPR
LGBMDF	0.9451	**0.9471**	**0.8924**	**0.9844**	**0.9857**
AOPEDF	**0.9463**	0.9447	0.8911	0.9842	0.9855
NEDTP	0.9194	0.9267	0.8462	0.9714	0.9690
SVM	0.8869	0.9286	0.8162	0.9668	0.9664
RF	0.9138	0.9348	0.8488	0.9784	0.9798

The bold values represent the maximum value of each estimator setting under each evaluation metric.

By comparing the Sn, Sp, MCC, AUC, AUPR and computation time of the 4 estimator combinations, we chose three ExtraTrees and three LightGBMs as estimators at each layer, and then utilized this cascade forest for DTI prediction. To demonstrate the merits of our model, we compared it with other four baseline models on the same dataset. After 5-fold cross-validation, we obtained the Sn, Sp, MCC, AUC and AUPR of the five models, the Sp (0.9471), MCC (0.8924), AUC (0.9844) and AUPR (0.9857) of LGBMDF were higher than AOPEDF, NEDTP, RF and SVM. The Sn (0.9451) was slightly inferior to AOPEDF, but higher than other three methods. Furthermore, the calculation time of LGBMDF was less than half of that of AOPEDF.

In summary, the method proposed in this paper shows higher prediction accuracy with the current state-of-the-art methods, and greatly improves the computational speed. We believe this will accelerate the drug development process to a certain extent. Certainly, there are still some shortcomings in this paper, such as feature extraction method. We believe that if there is a better way to extract features, the prediction accuracy will also be improved. Moreover, our method could also be applied in other studies, such as in exploring the link between microbes and cancer.

Data availability statement

The data and code for LGBMDF is available at https://github.com/TLanCZ/LGBMDF.

Author contributions

YP proposed the model and completed the manuscript writing. ZZ and XH assisted in completing the model construction. SZ and ZY reviewed and revised the manuscript. ZY provided financial support. All authors contributed to the article and approved the submitted version.

Funding

This research was supported by National Natural Science Foundation of China (no: 62072296).

Conflict of interest

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

Publisher's note

References

Al Daoud, E. (2019). Comparison between XGBoost, light GBM and cat boost using a home credit dataset. *Int. J. Comput. Inf. Eng.* 13, 6–10. doi: 10.5281/zenodo.3607805

An, Q., and Yu, L. (2021). A heterogeneous network embedding framework for predicting similarity-based drug-target interactions. *Brief. Bioinform.* 22:bbab275. doi: 10.1093/bib/bbab275

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., et al. (2004). Uni Prot: the universal protein knowledgebase. *Nucleic Acids Res.* 32, 115D–1119D. doi: 10.1093/nar/gkh131

Bagherian, M., Kim, R. B., Jiang, C., Sartor, M. A., Derksen, H., and Najarian, K. (2021). Coupled matrix–matrix and coupled tensor–matrix completion methods for predicting drug–target interactions. *Brief. Bioinform.* 22, 2161–2171. doi: 10.1093/bib/bbaa025

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Cao, D. S., Zhang, L. X., Tan, G. S., Xiang, Z., Zeng, W. B., Xu, Q. S., et al. (2014). Computational prediction of drug target interactions using chemical, biological, and network features. *Mol Inform* 33, 669–681. doi: 10.1002/minf.201400009

Chen, X., Liu, M.-X., and Yan, G.-Y. (2012). Drug–target interaction prediction by random walk on the heterogeneous network. *Mol. Bio Syst.* 8, 1970–1978. doi: 10.1039/c2mb00002d

Chen, C., Shi, H., Jiang, Z., Salhi, A., Chen, R., Cui, X., et al. (2021). DNN-DTIs: improved drug-target interactions prediction using XGBoost feature selection and deep neural network. *Comput. Biol. Med.* 136:104676. doi: 10.1016/j.compbiomed.2021.104676

Chen, M., and Yin, Z. (2022). Classification of Cardiotocography based on Apriori algorithm and multi-model ensemble classifier. *Front. Cell Dev. Biol.* 10:888859. doi: 10.3389/fcell.2022.888859

Cheng, F., Kovács, I. A., and Barabási, A.-L. (2019a). Network-based prediction of drug combinations. *Nat. Commun.* 10, 1–11.

Cheng, F., Lu, W., Liu, C., Fang, J., Hou, Y., Handy, D. E., et al. (2019b). A genome-wide positioning systems network algorithm for in silico drug repurposing. *Nat. Commun.* 10, 1–14. doi: 10.1038/s41467-019-10744-6

Chu, Y., Kaushik, A. C., Wang, X., Wang, W., Zhang, Y., Shan, X., et al. (2021a). DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features. *Brief. Bioinform.* 22, 451–462. doi: 10.1093/bib/bbz152

Chu, Y., Shan, X., Chen, T., Jiang, M., Wang, Y., Wang, Q., et al. (2021b). DTI-MLCD: predicting drug-target interactions using multi-label learning with community detection method. *Brief. Bioinform.* 22:bbaa205. doi: 10.1093/bib/bbaa205

Ding, Y., Tang, J., and Guo, F. (2017). Identification of drug-target interactions via multiple information integration. *Inf. Sci.* 418-419, 546–560. doi: 10.1016/j.ins.2017.08.045

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., et al. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* 40, D1100–D1107.

Gönen, M. (2012). Predicting drug–target interactions from chemical and genomic kernels using Bayesian matrix factorization. *Bioinformatics* 28, 2304–2310. doi: 10.1093/bioinformatics/bts360

Guo, F., Yin, Z., Zhou, K., and Li, J. (2021). PLncWX: a machine-learning algorithm for plant lncRNA identification based on WOA-XGBoost. *J. Chem.* 2021, 1–11. doi: 10.1155/2021/6256021

Hasan Mahmud, S. M., Chen, W., Jahan, H., Dai, B., Din, S. U., and Dzisoo, A. M. (2020). DeepACTION: a deep learning-based method for predicting novel drug-target interactions. *Anal. Biochem.* 610:113978. doi: 10.1016/j.ab.2020.113978

Hernandez-Boussard, T., Whirl-Carrillo, M., Hebert, J. M., Gong, L., Owen, R., Gong, M., et al. (2007). The pharmacogenetics and pharmacogenomics knowledge base: accentuating the knowledge. *Nucleic Acids Res.* 36, D913–D918. doi: 10.1093/nar/gkm1009

Jarada, T. N., Rokne, J. G., and Alhajj, R. (2021). SNF–CVAE: computational method to predict drug–disease interactions using similarity network fusion and collective variational autoencoder. *Knowl. Based Syst.* 212:106585. doi: 10.1016/j.knosys.2020.106585

Jin, S., Niu, Z., Jiang, C., Huang, W., Xia, F., Jin, X., et al. (2021). HeTDR: drug repositioning based on heterogeneous networks and text mining. *Patterns (N Y)* 2:100307. doi: 10.1016/j.patter.2021.100307

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). LightGBM: a highly efficient gradient boosting decision tree. *Adv. Neural Inf. Proces. Syst.* 30, 3149–3157.

Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., et al. (2014). DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Res.* 42, D1091–D1097. doi: 10.1093/nar/gkt1068

Li, Y., Liu, X. Z., You, Z. H., Li, L. P., Guo, J. X., and Wang, Z. (2020). A computational approach for predicting drug–target interactions from protein sequence and drug substructure fingerprint information. *Int. J. Intell. Syst.* 36, 593–609. doi: 10.1002/int.22332

Lihong, P., Wang, C., Tian, X., Zhou, L., and Li, K. (2021). Finding lncRNA-protein interactions based on deep learning with dual-net neural architecture. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 19, 3456–3468. doi: 10.1109/TCBB.2021.3116232

Lin, W., Wu, L., Zhang, Y., Wen, Y., Yan, B., Dai, C., et al. (2022). An enhanced cascade-based deep forest model for drug combination prediction. *Brief. Bioinform.* 23:bbab562. doi: 10.1093/bib/bbab562

Liu, T., Lin, Y., Wen, X., Jorissen, R. N., and Gilson, M. K. (2007). BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res.* 35, D198–D201. doi: 10.1093/nar/gkl999

Liu, Y., Wu, M., Miao, C., Zhao, P., and Li, X.-L. (2016). Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. *PLoS Comput. Biol.* 12:e1004760. doi: 10.1371/journal.pcbi.1004760

Mei, J. P., Kwoh, C. K., Yang, P., Li, X. L., and Zheng, J. (2013). Drug-target interaction prediction by learning from local information and neighbors. *Bioinformatics* 29, 238–245. doi: 10.1093/bioinformatics/bts670

Mousavian, Z., Khakabimamaghani, S., Kavousi, K., and Masoudi-Nejad, A. (2016). Drug-target interaction prediction from PSSM based evolutionary information. *J. Pharmacol. Toxicol. Methods* 78, 42–51. doi: 10.1016/j.vascn.2015.11.002

Olayan, R. S., Ashoor, H., and Bajic, V. B. (2018). DDR: efficient computational method to predict drug-target interactions using graph mining and machine learning approaches. *Bioinformatics* 34, 1164–1173. doi: 10.1093/bioinformatics/btx731

Pawson, A. J., Sharman, J. L., Benson, H. E., Faccenda, E., Alexander, S. P., Buneman, O. P., et al. (2014). The IUPHAR/BPS guide to PHARMACOLOGY: an expert-driven knowledgebase of drug targets and their ligands. *Nucleic Acids Res.* 42, D1098–D1106. doi: 10.1093/nar/gkt1143

Peng, J., Wang, Y., Guan, J., Li, J., Han, R., Hao, J., et al. (2021). An end-to-end heterogeneous graph representation learning-based framework for drug-target interaction prediction. *Brief. Bioinform.* 22:bbaa430. doi: 10.1093/bib/bbaa430

Peng, L., Wang, F., Wang, Z., Tan, J., Huang, L., Tian, X., et al. (2022). Cell–cell communication inference and analysis in the tumour microenvironments from single-cell transcriptomics: data resources and computational strategies. *Brief. Bioinform.* 23:bbac234. doi: 10.1093/bib/bbac234

Pliakos, K., Vens, C., and Tsoumakas, G. (2019). Predicting drug-target interactions with multi-label classification and label partitioning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 18, 1596–1607. doi: 10.1109/TCBB.2019.2951378

Pu, Y., Li, J., Tang, J., and Guo, F. (2021). DeepFusionDTA: drug-target binding affinity prediction with information fusion and hybrid deep-learning ensemble model. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 19, 2760–2769. doi: 10.1109/TCBB.2021.3103966

Sajadi, S. Z., Zare Chahooki, M. A., Gharaghani, S., and Abbasi, K. (2021). AutoDTI++: deep unsupervised learning for DTI prediction by autoencoders. *BMC Bioinformatics* 22:204. doi: 10.1186/s12859-021-04127-2

Shen, L., Liu, F., Huang, L., Liu, G., Zhou, L., and Peng, L. (2022). VDA-RWLRLS: An anti-SARS-CoV-2 drug prioritizing framework combining an unbalanced bi-random walk and Laplacian regularized least squares. *Comput. Biol. Med.* 140:105119. doi: 10.1016/j.compbiomed.2021.105119

Tanoori, B., Jahromi, M. Z., and Mansoori, E. G. (2021). Drug-target continuous binding affinity prediction using multiple sources of information. *Expert Syst. Appl.* 186:115810. doi: 10.1016/j.eswa.2021.115810

Vapnik, V. N., and Chervoneva, A. (1964). On class of perceptrons. *Autom. Remote. Control.* 25:103.

Wang, F., Lei, X., Liao, B., and Wu, F. X. (2022). Predicting drug-drug interactions by graph convolutional network with multi-kernel. *Brief. Bioinform.* 23:bbab511. doi: 10.1093/bib/bbab511

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., et al. (2018). DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 46, D1074–D1082. doi: 10.1093/nar/gkx1037

Yamanishi, Y., Kotera, M., Kanehisa, M., and Goto, S. (2010). Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics* 26, i246–i254. doi: 10.1093/bioinformatics/btq176

Yang, H., Qin, C., Li, Y. H., Tao, L., Zhou, J., Yu, C. Y., et al. (2016). Therapeutic target database update 2016: enriched resource for bench to clinical drug target and targeted pathway information. *Nucleic Acids Res.* 44, D1069–D1074. doi: 10.1093/nar/gkv1230

Yang, Z., Zhong, W., Zhao, L., and Yu-Chian Chen, C. (2022). MGraphDTA: deep multiscale graph neural network for explainable drug-target binding affinity prediction. *Chem. Sci.* 13, 816–833. doi: 10.1039/d1sc05180f

You, J., McLeod, R. D., and Hu, P. (2019). Predicting drug-target interaction network using deep learning model. *Comput. Biol. Chem.* 80, 90–101. doi: 10.1016/j.compbiolchem.2019.03.016

Yuan, Q., Gao, J., Wu, D., Zhang, S., Mamitsuka, H., and Zhu, S. (2016). DrugE-rank: improving drug–target interaction prediction of new candidate drugs or targets by ensemble learning to rank. *Bioinformatics* 32, i18–i27. doi: 10.1093/bioinformatics/btw244

Zeng, X., Zhu, S., Hou, Y., Zhang, P., Li, L., Li, J., et al. (2020a). Network-based prediction of drug-target interactions using an arbitrary-order proximity embedded deep forest. *Bioinformatics* 36, 2805–2812. doi: 10.1093/bioinformatics/btaa010

Zeng, X., Zhu, S., Lu, W., Liu, Z., Huang, J., Zhou, Y., et al. (2020b). Target identification among known drugs by deep learning from heterogeneous networks. *Chem. Sci.* 11, 1775–1797. doi: 10.1039/c9sc04336e

Zhan, X., You, Z., Yu, C., Li, L., and Pan, J. (2020). Ensemble learning prediction of drug-target interactions using GIST descriptor extracted from PSSM-based evolutionary information. *Biomed. Res. Int.* 2020, 4516250–4516210. doi: 10.1155/2020/4516250

Zhang, Z., Cui, P., Wang, X., Pei, J., Yao, X., and Zhu, W. (2018). "Arbitrary-Order Proximity Preserved Network Embedding", In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Zhang, Y., Jiang, Z., Chen, C., Wei, Q., Gu, H., and Yu, B. (2022). DeepStack-DTIs: predicting drug-target interactions using LightGBM feature selection and deep-stacked ensemble classifier. *Interdiscip. Sci.* 14, 311–330. doi: 10.1007/s12539-021-00488-7

Zhou, Z.-H., and Feng, J. (2017). "Deep Forest: Towards An Alternative to Deep Neural Networks", in: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 3553–3559.

Zhou, L., Li, Z., Yang, J., Tian, G., Liu, F., Wen, H., et al. (2019). Revealing drug-target interactions with computational models and algorithms. *Molecules* 24:1714. doi: 10.3390/molecules24091714

Zhou, L., Wang, Z., Tian, X., and Peng, L. (2021). LPI-deepGBDT: a multiple-layer deep framework based on gradient boosting decision trees for lncRNA–protein interaction identification. *BMC Bioinformatics* 22, 1–24. doi: 10.1186/s12859-021-04399-8

Zhou, K., Yin, Z., Peng, Y., and Zeng, Z. (2022). Methods for continuous blood pressure estimation using temporal convolutional neural networks and ensemble empirical mode decomposition. *Electronics* 11:1378. doi: 10.3390/electronics11091378