Predicting Drug-target Interaction via Wide and Deep Learning

Yingyi Du
School of Data and Computer Science,
Sun Yat-Sen University,
Guangzhou, China
duyy3@mail2.sysu.edu.cn

Jihong Wang
School of Data and Computer Science
Sun Yat-Sen University,
Guangzhou, China
redblue04@163.com

Xiaodan Wang
School of Pharmaceutical Chemistry and Chemical Engineering, Guangdong Pharmaceutical University,
Zhongshan, China
wangxiaodan@gdpu.edu.cn

Jiyun Chen
School of Data and Computer Science,
Sun Yat-Sen University,
Guangzhou, China
1211169311@qq.com

Huiyou Chang
School of Data and Computer Science, Sun Yat-Sen University
Guangzhou, China
isschy@mail.sysu.edu.cn

ABSTRACT

Identifying the interactions of approval drugs and targets is essential in medicine field, which can facilitate the discovery and reposition of drugs. Due to the tendency towards machine learning, a growing number of computational methods have been applied to the prediction of the drug-target interactions (DTIs). In this paper, we propose a wide and deep learning framework combining a generalized linear model and a deep feed-forward neural network to address the challenge of predicting the DTIs precisely. The proposed method is a joint training of the wide and deep models, which is implemented by feeding the weighted sum of the results obtained from the wide and deep models into a logistic loss function using mini-batch stochastic gradient descent. The results of this experiment indicate that the proposed method increases the accuracy of prediction for DTIs, which is superior to other methods.

CCS Concepts

• **Applied computing → Bioinformatics**

Keywords

drug-target interaction prediction; wide and deep model; machine learning; deep learning; DrugBank.

1. INTRODUCTION

Drug target is biological macromolecule in the body whose activity can be modified by drugs, resulting in a curative effect on the diseases. Identifying the drug-target interactions (DTIs) is not only important to clarify the mechanism of drug, but also the first step in the discovery of a medicine [1]. However, experimental identification is expensive and time-consuming with side effects. As an effective complement of the experimental method, the computational method of prediction of DTIs provides an efficient and low-risk approach, which can play a significant role in predicting targets and saving costs [2].

Traditional methods for predicting DTIs previously include docking simulation [3][4] and ligand-based approaches [5][6]. Docking simulation needs the 3D structure of the target protein, which is often unavailable, because determining 3D structure of most proteins is complicated, especially GPCRs. As well as, the ligand-based methods rely on the number of ligands associated with target proteins. Kinds of chemogenomic approaches have been proposed to avoid these problems.

Chemogenomic approaches, using genomic and chemical property, can be classified as learning-based methods, graph-based methods [7][8] and network-based methods [9][10]. In recent years, a number of studies have explored the benefits and effects of machine learning methods in bioinformatics field. Learning-based approaches, like supervised learning methods, are the *in-silico* approaches which predict unknown DTIs according to the experimentally validated DTIs, along with chemical and genomic features. Deng et al. [11] explored an automatic presented support vector machine (SVM) method for prediction of drug protein interactions, and also used an automatic procedure to randomly select negative datasets to deal with the data imbalance problem. Rayhan et al. [12] performed the adaptive boosting algorithm (AdaBoost) to identify the DTIs using the decision trees as the weak classifiers; they modified under-sampling method based on clustering and proposed the method so-called *iDTI-ESBoost* integrating evolutionary and structural features with boosting.

As deep learning has attracted a lot of attention in different fields, some researchers applied the deep learning methods to the biology and chemistry fields. One of the advantages of the deep learning models over traditional machine learning is the lightweight feature engineering they require [13]. Wang et al. [14] proposed a predicting method of deep learning for drug-target interactions by using the stacked auto-encoder to learn the high-level features, which has mined the hidden information from raw data automatically; then fed the features containing the extracted features and drug molecule fingerprints to a rotation forest classifier to predict the DTIs. Wen et al. [15] developed a deep-

ICBCB 2018, March 12–14, 2018, Chengdu, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6348-8/18/03…$15.00
https://doi.org/10.1145/3194480.3194491

learning framework, *DeepDTIs*, based on deep belief network (DBN) to predict new DTIs between the approved drugs and targets, which abstracted feature representation from raw data using unsupervised pre-training; considering the situation that one target protein and another protein which belongs to different categories can work with the same drug, they used a global data set, rather than the golden standard datasets.

It is commonly believed that the ensemble method always outperforms any of its base learners. Zhang et al. [16] proposed a random projection ensemble approach, *DrugRPE*, which was an ensemble REPTree system built by several random projections; they projected the feature descriptors onto a reduced space using random projection to overcome the problem of the feature vectors with uncertain length, thus the dimensions of the feature vectors were fixed. Cao et al. [17] used random forests (RFs) to develop a computational framework for predicting drug-target interactions by integrating chemical, biological, and network features. This method which integrated three types of information showed that the additional network information appropriately improved the performance of their model. Ali Ezzat et al. [18] provided a solution to the imbalance data problem via an ensemble learning method, which used the decision trees as base learners; they performed random sampling method and clustering to alleviate the within and between class imbalance problem. It is a challenge to combine multi-type information which leads to the over-fitting problem and a high time complexity. Moreover, most ensemble approaches are based on decision trees which are not capable of processing large ID data and will lose valuable information.

In this paper, we develop a learning framework combined wide and deep models to overcome the problem of predicting DTIs by jointly training a linear model and a neural network. Some previous computational approaches used the golden standard datasets [19], which involve four types of target proteins, namely, enzymes, ion channels, g-protein coupled receptors and nuclear receptors. Using four types of target features will miss the interactions between distinct kinds of targets for the giving drug. As well as, each category will train a model leading to the increased time and storage cost. To ensure the completeness and accuracy of the results, we use a global dataset, *DrugBank*, in the experiment, which has 11099 drug-target interactions labeled as positive. Meanwhile, we use an opensource python package, *PyDPI* [20], to compute the molecular descriptors of drug molecules and the physicochemical features of proteins from amino acid sequences. The proposed wide and deep learning method combines the advantages of both wide model and deep model. The wide model is expert in recognizing relevance from existing training data, but lacks generalization, usually requiring more feature engineering. In contrast, the deep model is easy to over-generalize. Training the wide and deep models jointly can not only combine the advantages of both models, but also overcomes the shortcomings of the two respectively. To the best of our knowledge, it is the first time to predict DTIs by the wide and deep method. The experimental results indicate that the proposed method is superior to other state-of-the-art methods on the same datasets.

2. MATERIALS AND METHODS

2.1 Datasets

The positive drug-target interactions data was obtained from *DrugBank* database [21]. We labeled 10731 positive drug-target interactions with 1, which means there is interaction between the drug-target pair. One of the difficulties in prediction of DTIs is that the number of uncertain data is much greater than known interactions. To mitigate the imbalanced data problem, we used random down-sampling to generate 22714 negative samples by following steps [22]. First, each drug-target pair was divided into separate drug and target to form a drug set and a target set. The number of drugs and targets is 4292 and 2311 respectively. Second, we randomly selected drugs and targets from the drug set and target set to construct new drug-target pairs as negative samples. Third, drug-target pairs occurring in the positive set were replaced by other non-redundant pairs. Finally, these pairs of drug-target were labeled with 0, thus, we generated a negative dataset containing 22719 samples which are about twice as the positive set.

We used an opensource python package, *PyDPI*, to extract the feature vectors of drug molecules and target proteins. Molecular descriptors are used to represent the features of small chemical molecules, which are regarded as a numerical representation of the chemical features of drugs. Analogously, protein descriptors are numerical representations of the protein features. *PyDPI* provides a download module to automatically download the amino acid sequences and chemical molecular structures [20]. We calculated the protein descriptors with the *PyPro* module, including 20 amino acid compositions, 147 CTDs, 400 dipeptide compositions, 240 Moran autocorrelations, 240 normalized Moreau-Broto autocorrelations, 40 amphiphilic pseudo amino acid compositions, 100 quasi-sequence order descriptors, 512 conjoint triad features, 90 sequence order coupling numbers and 30 pseudo amino acid compositions. We calculated the drug molecule descriptors with the *PyDrug* module, including 30 molecular constitutional descriptors, 25 topological descriptors, 44 molecular connectivity indices, 316 E-state descriptors, 7 kappa shape descriptors, 60 MOE-type descriptors, 32 Geary autocorrelations, 32 Moran autocorrelations, 32 Moreau-Broto autocorrelations, 25 charge descriptors and 6 molecular properties. The feature vector of the DTI, f = (f_1, f_2, ..., f_k), is constructed by connecting the drug molecular descriptor vector fd = (fd_1, fd_2, ..., fd_m) and the protein descriptor vector ft = (ft_1, ft_2, ..., ft_n), where f = (fd, ft) with the size of k (= m + n, (m is the size of fd, n is the size of ft)). The dimension of the drug molecular descriptor is m (= 609) and the target descriptor is n (= 1819). Totally, each drug-target pair has 2430 features, counting in *drugID* and *targetID*.

2.2 Data Preprocessing

There is incomplete and inconsistent dirty-data in the data we acquired, which cannot be applied to the model directly. To improve the quality of data mining, the data has been preprocessed as follows:

- *Missing data*. We dropped out 44 instances containing missing data directly, because the dimension of missing data (2428) is much more than the number of instances with missing data (44). The size of final dataset is 33449 rows × 2431 columns.
- *NaN and inf*. There are some *Not a Number* (*NaN*) and *infinity* (*inf*) in the features extracted by *PyDPI*. The model cannot process these types of data, so we filled them with average.
- *Constant column*. There is only one value in each constant column. These features are useless for boosting result, therefore, we dropped them out.

– *Normalization*. Normalization can avoid numerical problems and convert different evaluation criteria of data to the same one by z-score function expressed as,

$$x^* = \frac{x-\mu}{\sigma}$$

where x is a specific value, μ is the average of the column, σ is the standard deviation, and x^*represents the distance between the original score and the average.

2.3 Wide-and-deep Model

The training set includes n instances (x, y), where $x = [drug_{id}, target_{id}, x_1, x_2, \dots, x_k]$ is a (k+2)-dimension vector involving a pair of drug and target, categorical features and continuous features, and y ∈ (0,1) is the label indicating whether there is interaction between the drug-target pair (y=1 means there is interaction between drug and target, and y=0 otherwise). The wide-and-deep model is a combination of linear model and DNN model to model low-and-high-dimension features. It optimizes the parameters of the two models simultaneously during the training phase, so that the whole model has a better ability in predicting DTIs than any other two single models [23].

The wide model is a generalized linear model. The input features of wide part are continuous features, sparse categorical features and transformed features. The transformation of the raw data here is *cross-product*, which build higher-dimensional discrete features. In linear model, each feature is multiplied by the weight matrix to obtain the score of feature, then add up scores and bias to obtain the result of wide part, which can be expressed as

$$y_{wide} = w^T x + b \ ,$$

where y_{wide} is the result of wide part, *x* is the vector of features with expertise feature engineering, *w* is the weights of *x*, and *b* is the bias.

The deep model is a feed-forward neural network with an input layer, some hidden layers and an output layer. Each feature will be converted to a low-dimensional and real-valued vector referred to as *embedding vector*. After constructing the input layer with real-value and embedding vectors, hidden layers are built from bottom to top using the results of the input layer or the previous hidden layer. The output layer is fully connected with the last hidden layer. Each hidden layer performs

$$a^{(l+1)} = f(W^{(l)} a^{(l)} + b^{(l)}) \ ,$$

where *l* is the number of layers, *a* is the activations of the l-th layer, *W* and *b* are weights and bias at the l-th layer respectively. *f* is the activation function, here rectified linear units (ReLUs).

As illustrated in figure 1, wide-and-deep model consists of two parts, wide component and deep component. The prediction of model is:

$$P(Y = 1|x) = \sigma(w_{wide}^T [x, \phi(x)] + w_{deep}^T a^{l_f} + b)$$

where *Y=1* means the interaction between the given drug-target pair is positive, $\sigma(\cdot)$ is sigmoid function to transform the result into probability, w_{wide} is the weight vectors of the wide model, $w_{deep}^T a^{l_f}$is the final activations, a^{l_f} and their weight vectors of the deep model, b is the bias, $\phi(x)$ is the *cross product* of the input *x*.

During the training phase, the combined model feeds the weighted sum of the results obtained from the wide model and deep model into a logistic loss function. Then feed the training loss back to the linear optimizer and DNN optimizer respectively. Using follow-the-regularized-leader (*FTRL*) algorithm with L_1 regularization can converge into a valid combination of features quickly during the training of linear model. The DNN component adjusts the weights of hidden layers by back-propagation, and updates the embedding vector. The joint training feeds the loss back to linear and deep part simultaneously to update parameters. Compared with other ensemble learning methods, joint training combines the models during training, rather than combining the models during final predicting phase. The parameters updated through single models will cause influence on the training errors for both wide and deep part. We used 10-fold cross validation as the method of testing the accuracy of the algorithm to obtain a reliable and stable model.

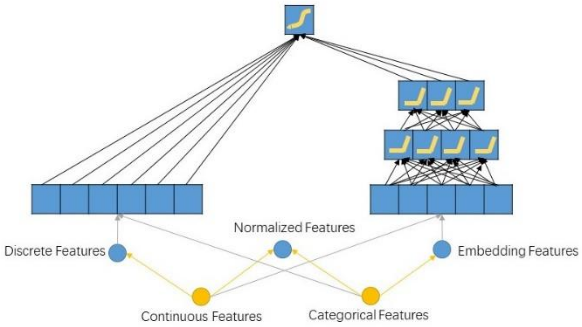

Figure 1. The architecture of wide-and-deep model.

3. RESULTS AND DISCUSSION

3.1 Performance Evaluation

As mentioned previously, the data distribution is not balanced. In this case, we utilized four types of evaluation methods to reflect the experimental results and the superiority of the supposed method compared with other four algorithms objectively. The evaluation includes accuracy (*ACC*), area under ROC curve(*AUC*), precision(*PRE*), sensitivity(*SEN*), defined as,

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$PRE = \frac{TP}{TP + FP}$$

$$SEN = \frac{TP}{TP + FN}$$

$$AUC = \frac{\sum_{i \in positiveClass} rank_i - \frac{M(1+M)}{2}}{M \times N}$$

where *M* is the number of positive instances and *N* is the negatives, *rank* is the position of the sorted positive instances. *TP* is true positive that indicates the number of samples which are correctly divided into the positives. *FP* is false positive that indicates the number of samples which are falsely divided into the positives. *FN* is false negative that indicates the number of samples which are falsely divided into the negatives. *TN* is true negatives that indicates the number of samples which are correctly divided into the negatives. In general, the higher the accuracy is, the better the predicting capability is. The sensitivity represents the proportion of records predicted correctly in all positive records, which measures the ability of classifier to identify positive records. The precision represents the proportion of the truly positive records in the records divided into positives.

3.2 Parameter Settings

In this work, the proposed predicting model was constructed by *tensorflow* and *sklearn* python package. The wide-and-deep model has four parameters need to be set : linear_feature_colum , dnn_feature_columns, the number of hidden layers (*nhl*) and the number of units of each layer (*nu*), which need to be further optimized. The *linear_feature_columns* and *dnn_feature_columns* are the feature columns fed into the wide model and deep model, including raw feature columns and transformed feature columns. For wide part, feature columns involve raw categorical features (*cat*), raw continuous features (*con*), *cross-product* (*crs*) and discrete features (*dis*), while deep part involves raw continuous features (*con*), embedding features(*emb*) and normalized features(*norm*). The numbers of hidden layers and units are the determining factors of the architectures of deep model. To achieve the better performance, we used *grid search* to screen *nhl* and *nu* values ranging from [50,25] to [500,500] with a step of 50. The optimized [*nhl*, *nu*] is [500,250].

3.3 Performance of Prediction on DTIs

In this experiment, the accuracy of the algorithm is tested with 10-fold cross validation, which means that the dataset is randomly divided into 10 subsets, 9 of which are used as training data and one as test data. The above test procedure is repeated 10 times to get the corresponding correct rate. The 10 results can be averaged to bring about the final accuracy.

Table 1. Performance comparison of our method under different feature engineering

	ACC	AUC	PRE	SEN
cat+con+emb	0.6110	0.5708	0.5454	0.4531
cat+con+emb+dis	0.7699	0.7822	0.7392	0.4577
cat+con+emb+dis+norm	0.8574(±0.16)	0.8917(±0.02)	0.7964(±0.05)	0.7539(±0.03)

Using the optimized values of the numbers of hidden layers and units per layer, several predicting wide-and-deep models were constructed based on different feature columns. The Table 1 shows the result from the different combination of feature columns, from which can be seen that the proposed model constructed by *sps+con+emb+dis+norm* obtained significant improvement statistically. By the way, the *cross-product* columns lead to dimension explosion, so the result is hard to obtain. As shown in Table 1, the accuracy, AUC, precision and sensitivity of the proposed method constructed by categorical, continuous, embedding, discrete and normalized features are 0.8574(±0.16), 0.8917(±0.02), 0.7964(±0.05) and 0.7539(±0.03), respectively. The results decrease by 0.0875, 0.1095, 0.0572 and 0.2962 respectively without normalized feature columns. Moreover, without discrete feature columns, the results decrease by another 0.1589, 0.2114, 0.1938, 0.0046. It is because of the components of the proposed method. The wide part of the model is a linear model, which requires more feature engineering to result in a better performance. However, too much feature engineering makes the model complicated and is easy to over-fit.

3.4 Comparison with Other Approaches

In order to show the superiority of the proposed approach, we built classification models using logistic regression (*LR*), random forests (*RF*), support vector machine (*SVM*) and deep neural network (*DNN*) respectively based on the same datasets. The results of comparison are listed in table 2. The accuracies of *LR*, *RF*, *DNN*, *SVM* and *wide-and-deep learning* are 0.7488, 0.7557, 0.8142, 0.7926 and 0.8574, respectively, in which our method reaches the highest value 0.8574 and the lowest is only 0.7488 from *LR*. The AUCs of five models are 0.8284, 0.8572, 0.8538, 0.8214 and 0.8917, respectively, which our method reaches the highest value, 0.8917 and the lowest reaches 0.8214. The precisions are 0.7526, 0.8769, 0.719080, 0.7329, 0.7964, respectively. The sensitivities are 0.5418, 0.6916, 0.6775, 0.5555 and 0.7539, respectively. As shown in Table 2, wide-and-deep model performs better than *LR*, *SVM* and *DNN* in *ACC*, *AUC*, *PRE* and *REC*. The precision of the proposed methods is lower than RF's. The precision is the proportion of the true positive in the records divided into the positives. In the problem of predicting DTIs, the number of positive instances is much fewer than the negatives and it is more significative to find the true positives, therefor *SEN* is more important between precision and sensitivity. Compared with single wide model and single deep model, the results from Table 2 approve our hypothesis that joint-training wide-and-deep model combines the advantages of the memorization of the wide component and the generalization of the deep component. Wide linear component can memorize sparse feature interactions effectively, while deep component can learn the unseen interactions previously through embedding features. The wide-and-deep model expands the predicting capacity through learning linear interactions between sparse features via linear model and highly nonlinear interactions between embeddings via deep neural networks. Through joint training, the weight of a single component will be affected by the training errors from both linear and deep models. Therefore, the wide component only needs focus on memorization through feature engineering, while deep component pays attention to generalization by embedding vectors. The overall performance of the proposed model has been appropriately improved.

Table 2. Performance comparison of our method with four methods on the same datasets

	ACC	AUC	PRE	SEN
LR	0.7488	0.8284	0.7526	0.5418
RF	0.7557	0.8572	**0.8769**	0.6916
DNN	0.8142	0.8538	0.7190	0.6775
SVM	0.7926	0.8214	0.7329	0.5555
Our method	**0.8574**	**0.8917**	0.7964	**0.7539**

4. CONCLUSIONS

Identification of the interactions between drugs and targets is key in drug discovery. In this paper, we proposed wide-and-deep learning, a novel computational method to predict drug-target interactions. The proposed method trains linear and non-linear model jointly based on a wide and deep learning framework, which combines the advantages of linear model and deep neural networks. We conducted experiments on a global dataset to compare the effectiveness of wide-and-deep model and the other models. The experiment results demonstrate that our proposed method is superior to others on the prediction of drug-target interactions. That is because it learns both highly nonlinear feature interactions and linear interactions. We believe that the proposed method will contribute to the process of drug discovery, also can be extended to find the new use of the 'old drugs'. Another interesting direction for further study is exploring the representations of protein sequence using wide-and-deep model to predict DTIs.

5. ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed. The work was supported by the Science and Technology Program of Guangzhou, China (62000-42050043).

6. REFERENCES

[1] Chen, X., Yan, C. C., Zhang, X., Zhang, X., Dai, F., Yin, J., and Zhang, Y. Drug-target interaction prediction: databases, web servers and computational models. *Briefings in Bioinformatics 17*, 4 (2016), 696.

[2] Nascimento, A. C. A., Prud Encio, R. B. C., and Costa, I. G. A multiple kernel learning algorithm for drug-target interaction prediction. *Bmc Bioinformatics 17*, 1 (2016), 46.

[3] Cheng, A. C., Coleman, R. G., Smyth, K. T., Cao, Q., Soulard, P., Caffrey, D. R., Salzberg, A. C., and Huang, E. S. Structure-based maximal affinity model predicts small-molecule druggability. *Nature Biotechnology 25*, 1 (2007), 71.

[4] Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., and Olson, A. J. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry 30*, 16 (2009), 2785–2791.

[5] Campillos, M., Kuhn, M., Gavin, A.-C., Jensen, L. J., and Bork, P. Drug target identification using side-effect similarity. *Science 321*, 5886 (2008), 263–266.

[6] Keiser, M. J., Roth, B. L., Armbruster, B. N., Ernsberger, P., Irwin, J. J., and Shoichet, B. K. Relating protein pharmacology by ligand chemistry. *Nature biotechnology 25*, 2 (2007).

[7] Chen, X., Liu, M.-X., and Yan, G.-Y. Drug–target interaction prediction by random walk on the heterogeneous network. *Molecular BioSystems 8*, 7 (2012), 1970–1978.

[8] Jaeger, S., Min, J., Nigsch, F., Camargo, M., Hutz, J., Cornett, A., Cleaver, S., Buckler, A., and Jenkins, J. L. Causal network models for predicting compound targets and driving pathways in cancer. *Journal of Biomolecular Screening 19*, 5 (2014), 791.

[9] Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., Zhou, W., Huang, J., and Tang, Y. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS computational biology 8*, 5 (2012), e1002503.

[10] Cheng, T., Hao, M., Takeda, T., Bryant, S. H., and Wang, Y. Large-scale prediction of drug-target interaction: a data-centric review. *Aaps Journal 19*, 5 (2017), 1264–1275.

[11] Deng N Y, Wang Y, and Yang Z X. Computationally Probing Drug-Protein Interactions Via Support Vector Machine. *Letters in Drug Design & Discovery 7*,5(2010):-.

[12] Rayhan, F., Ahmed, S., Shatabda, S., Farid, D. M., Mousavian, Z., Dehzangi, A., and Rahman, M. S. idti-esboost: Identification of drug target interaction using evolutionary and structural features with boosting. *arXiv preprint arXiv:1707.00994* (2017).

[13] Deng, L., Yu, D., et al. Deep learning: methods and applications. *Foundations and Trends in Signal Processing 7*, 3–4 (2014), 197–387.

[14] Wang, L., You, Z.-H., Chen, X., Xia, S.-X., Liu, F., Yan, X., and Zhou, Y. Computational methods for the prediction of drug-target interactions from drug fingerprints and protein sequences by stacked auto-encoder deep neural network. In *International Symposium on Bioinformatics Research and Applications* (2017), Springer, pp. 46–58.

[15] Wen, M., Zhang, Z., Niu, S., Sha, H., Yang, R., Yun, Y., and Lu, H. Deep-learning-based drug–target interaction prediction. *Journal of Proteome Research 16*, 4 (2017), 1401–1409.

[16] Zhang, J., Zhu, M., Chen, P., and Wang, B. Drugrpe: Random projection ensemble approach to drug-target interaction prediction. *Neurocomputing 228* (2017), 256–26

[17] Cao, D. S., Zhang, L. X., Tan, G. S., Xiang, Z., Zeng, W. B., Xu, Q. S., and Chen, A. F. Computational prediction of dru-target interactions using chemical, biological, and network features. *Molecular Informatics 33*, 10 (2014), 669.

[18] Ezzat, A., Wu, M., Li, X.-L., and Kwoh, C.-K. Drug-target interaction prediction via class imbalance-aware ensemble learning. *BMC bioinformatics 17*, 19 (2016), 509.

[19] Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., and Kanehisa, M. Prediction of drug target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics 24*, 13 (2008), i232–i240.

[20] Cao, D. S., Liang, Y. Z., Yan, J., Tan, G. S., Xu, Q. S., and Liu, S. Pydpi: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. *Journal of Chemical Information & Modeling 53*, 11 (2013), 3086.

[21] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., and Neveu, V. Drugbank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research 42*, Database issue (2014), D1091.

[22] He, Z., Zhang, J., Shi, X. H., Hu, L. L., Kong, X., Cai, Y. D., and Chou, K. C. Predicting drug-target interaction networks based on functional groups and biological features. *Plos One 5*, 3 (2010), e9603.

[23] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (2016), ACM, pp. 7–10.