Analytical Biochemistry 610 (2020) 113978

Contents lists available at ScienceDirect

Analytical Biochemistry

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

DeepACTION: A deep learning-based method for predicting novel drug-target interactions

S.M. Hasan Mahmud [a], Wenyu Chen [a,*], Hosney Jahan [b], Bo Dai [a], Salah Ud Din [a], Anthony Mackitz Dzisoo [c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[b] College of Computer Science, Sichuan University, Chengdu, 610065, China
[c] Center for Informational Biology, University of Electronic Science and Technology of China, Chengdu, 611731, China

ARTICLE INFO

Keywords:
Drug-target interaction
Convolutional neural network
Data balancing
Feature extraction
LASSO

ABSTRACT

Drug-target interactions (DTIs) play a key role in drug development and discovery processes. Wet lab prediction of DTIs is time-consuming, expensive, and tedious. Fortunately, computational approaches can identify new interactions (drug-target pairs) and accelerate the process of drug repurposing. However, a vast number of interactions remain undiscovered; therefore, we proposed a deep learning-based method (deepACTION) for predicting potential or unknown DTIs. Here, each drug chemical structure and protein sequence are transformed according to structural and sequence information using different descriptors to represent their features correctly. There have been some challenges, such as the high dimensionality and class imbalance of data during the prediction process. To address these problems, we developed the MMIB technique to balance the majority and minority instances in the dataset and utilized a LASSO model to handle the high dimensionality of the data. In addition, we trained the convolutional neural network algorithm with balanced and reduced features for accurate prediction of DTIs. In this study, the AUC is considered a primary evaluation metric for comparing the performance of the deep ACTION model with that of existing methods by a 5-fold cross-validation test. Our experiential dataset obtained from the DrugBank database and our deepACTION model achieved an AUC of 0.9836 for this dataset. The experimental results ensured that the model can predict significant numbers of new DTIs and provide complete information to motivate scientists to develop drugs.

1. Introduction

The determination of DTIs is a large research area that plays a vital role in discovering novel target proteins for known drugs or developing new drugs for existing targets [1,2]. A huge effort has been undertaken to detect new DTIs by wet laboratories; nevertheless, large numbers of possible interactions are untraced. Generally, few drugs receive clearance to reach the marketplace, where most drugs are refused within clinical trials due to high toxicity or side effects. Every step of wet-laboratory experiments to predict interactions is laborious and expensive; therefore, it is encouraged to develop computational (e.g., machine learning) approaches to detect new DTIs. Currently, the effort to determine DTIs is being reinforced by various computational tools with various types of datasets [3]. Based on the different chemicals and biological data sources, similar public databases focusing on verified interaction information have been developed, such as KEGG [4], DrugBank [5], ChEMBL [6], STITCH [7] and TTD [8]. These data sources store and offer web-laboratory-based experimented information that is suitable for creating a computational approach for determining new DTIs and are used as the gold standard dataset.

Existing computational approaches can be classified into three categories: ligand-based methods, docking-based methods, and chemogenomic methods. First, ligand-based methods exploit target protein similarity to predict interactions between drug chemical structures and protein sequences in a QSAR frame [9]. For example, Keiser et al. developed a similarity-based method to predict undiscovered protein targets from the chemical compound similarity of drugs where a ligand topology network was used to calculate the similarity point of every single set [10]. Similarly, Campillos et al. [11] exploited similarities of side effects to attain molecular activity between drugs and target

* Corresponding author. Computational Intelligence Laboratory (CIL), School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China.
E-mail address: cwy@uestc.edu.cn (W. Chen).

https://doi.org/10.1016/j.ab.2020.113978
Received 4 January 2020; Received in revised form 23 September 2020; Accepted 25 September 2020
Available online 6 October 2020

proteins. However, these methods produce poor results because of the inadequate known (interacting) ligands of proteins. Second, docking-based methods utilize dynamic simulation of the target protein to identify new unknown interactions. These methods are a potential technique, but they require the 3D structure of proteins for processing the prediction task. Most importantly, discovering the 3D structure of the target protein is a complex task that is done via a difficult experimental process. Therefore, it is time to develop efficient machine learning techniques based on the FASTA format instead of 3D structures of proteins, to identify new interactions (drug-target).

Currently, chemogenomic methods [12] have become very popular for identifying DTIs using machine learning techniques. These methods represent the datasets as positive and negative samples with the chemical structures and sequences of the drug-target to train a model classifier (algorithm) and afterward detect unknown interactions using the trained model. In a recent review paper [13], machine learning techniques are classified as similarity-based or feature-based methods [14, 15]. Similarity-based methods such as matrix factorization [16], kernel-based methods [17,18], and graph-based methods [19] have been proposed to address the prediction task on different types of data, whereas these methods process the input data as a feature vector and then label the drug-target pairs by binary value to indicate whether a pair has interacted or not between drugs and targets. Wang et al. [20] introduced a model (supervised learning) to predict DTIs using SVM, where authors developed a similarity-based technique for calculating protein-protein and drug-drug similarity scores. He et al. [15] developed a computational method to encode features from the biological properties of a target sequence and physiochemical features of the drug chemical structure. A unified space technique was first introduced by Yamanishi et al. [21] for extracting drug and protein features using a bipartite graph. The aim of this method is to predict potential pairs from the pharmacological effects of drug targets into an integrated framework. Recently, a boosting-based method called iDTi-CSsmoteB was introduced [2], which utilized undersampling and oversampling techniques to manage the imbalanced datasets. Moreover, this method applied different extraction techniques to generate evolutionary, sequence information and structural properties of input data. Another model was proposed by the same author [1], where a new cluster undersampling technique and a dimensionality reduction technique were developed to obtain accurate predictions from balanced and reduced features of drugs and targets. An ensemble model was proposed to effectively handle the imbalance issue. Huang et al. [22] developed a randomized tree method to identify DTIs. A substructure fingerprint is used to represent the drug structures, and the protein sequence is encoded by Pseudo-SMR.

A few studies recruited deep learning approaches (e.g., DNN) to predict DTIs (binary classification problem) using various input features of drugs and proteins in the drug discovery area, and these approaches may be able to handle the limitations of the existing techniques [23–25]. Some studies have successfully applied deep-belief networks [26] and stacked autoencoders [27] for the same purpose. Moreover, the stacked autoencoder technique combined with CNNs (Convolutional Neural Networks) and RNNs (Recurrent Neural Networks) represents genomic and chemical structures as numerical vector forms. Pahikkala et al. [28] introduced a KronRLS algorithm that uses Smith-Waterman and 2D compound similarity for expressing the drugs and proteins. Recently, the SimBoost technique has been developed to identify binding affinity with a boosting algorithm by applying the feature extraction process to represent DTIs [29]. The authors employed similarity information of the drug and target as well as feature information that was generated from the interaction network of pairs. In both studies, the authors utilized traditional ML algorithms and applied 2D structures of DT to obtain similarity information. Another method, DeepDTI, was proposed [30] using DBN (deep belief network) with an ECFP fingerprint for drugs and tripeptides, dipeptides and amino acids for proteins. Those authors also mentioned how the DL method is used for nonlinear feature combinations to minimize the limitations of ordinary descriptors from the performance of every single layer. Few existing methods consider gold standard datasets (Nuclear Receptors (NR), GPCRs (G), Ion Channels (I), and Enzymes (E)) with deep learning technology for predicting DTIs. Wang et al. [27] proposed a stacked autoencoder of a deep learning algorithm to adequately extract raw information from drug-protein features, where thirty protein sequences were transferred to PSSM features and drugs were transferred to substructure fingerprints. Their methods have the benefit of automatically attaining hidden information of protein sequences and generating useful features through iterations of multiple layers. Afterward, an ensemble classifier is used to predict the DTIs from the drug and protein features. Another author used the same datasets [31] and presented a novel CNN-based prediction system to identify DTIs with a new negative instance. Wang et al. [32] proposed an underlying but effective recurrent neural network (RNN) that uses Legendre Moment (LM) and PSSM with molecular substructure fingerprints (MSF) to derive complementary and shared information for predicting DTIs. Last, they applied Sparse PCA to reduce the drug-target features into a uniform vector form and constructed the DeepLSTM method for prediction. Another technique, the CNN-based DTI method (similar topology to LeNet-5 networks), was proposed [33] that combines PaDEL-Descriptors and Moran autocorrelation descriptors to construct a prediction framework for providing a complete function through convolutional layers, two pooling layers, and one fully connected layer.

The computational identification of DTIs is also a challenging task in the area of drug repurposing, which relies on a few aspects: (i) the amount of chemogenomic resources (drugs and proteins) is increasing quickly; and (ii) the number of known DTIs is less. Therefore, it is complicated to choose negative pairs from the datasets because there have been no verified negative pairs in public databases. To date, many computational approaches have been developed to accurately predict new interactions, but a large number of interactions are still undiscovered. Additionally, most of the models suffer high false-positive problems in the prediction stage, introducing biologically interpretable errors.

To tackle these limitations, we proposed a CNN-based model called deepACTION to identify potential DTIs using the chemical structure of drugs and sequence information of proteins. The drug-target pairs were downloaded from the DrugBank [5] database, and their respective chemical structures and protein sequences were also collected from the KEGG [4] database. First, the drug chemical structure is transformed into a topological, constitutional, and geometrical form, which completely represents the molecular properties. For a target sequence, different protein descriptors, such as dipeptide composition, amino acid composition (AAC), pseudoamino acid composition (PAAC), autocorrelation, quasi-sequence-order, and CTD, have been utilized to describe its sequence information. Subsequently, these extracted drug-protein features are combined to create valid data where interacting features (drug-target) are indicated as positive pairs and noninteracting features are indicated as negative pairs. Moreover, a data balancing technique is developed to manage the imbalanced drug-target dataset, and a feature selection model is also utilized to reduce the dimensionality of the features. Finally, the training model is constructed using the CNN classifier on the balanced and reduced features to predict the interacting and noninteracting pairs from the experimental dataset. As a result, the proposed model shows the highest performance compared to related classifiers and methods. We believe that our newly developed deepACTION model shows the effectiveness of feature extraction, data balancing, feature selection and classifiers that would encourage practitioners and researchers to use it to predict DTIs. Our proposed deepACTION model is presented in Fig. 1.

2. DTI problem formulation

Prediction in the DTI network for a dataset is devised by a bipartite

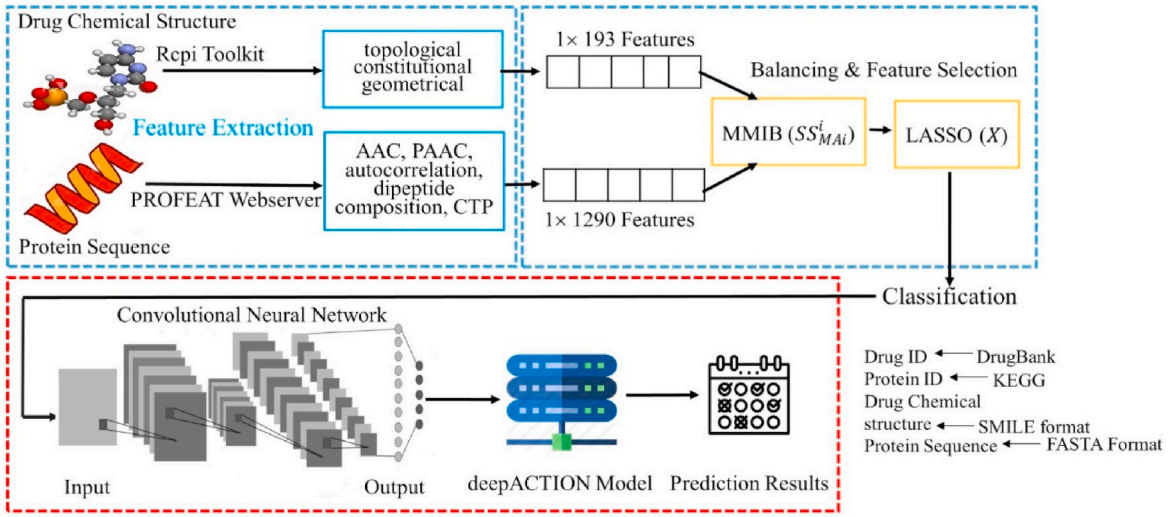

Fig. 1. DeepACTION has four main parts: feature extraction, data balancing, feature selection, and classification model. The feature extraction method is based on different drug and protein feature generation techniques from the DrugBank Dataset. The balancing technique MMIB is applied to the extracted features, and the LASSO technique is also utilized on balanced features. Then, the CNNs are used to train our model for predicting new interactions from the obtained features.

graph, $G = (V, E)$, where $V = D \cup T$ represents the set of vertices such that $D = \{d_1, d_2, d_3 \ldots d_m\}$represents the drugs of the dataset, $T = \{t_1, t_2, t_3 \ldots t_n\}$represents the targets of a dataset, and E indicates the set of edges among drugs and target proteins. Here, each $e = (d, t) \in E$ indicates the interaction between a drug and a target. The known interacting edges in the graph are considered positive instances, and the nonexistent (unknown) edges are considered negative instances. For example, Fig. 2 shows a bipartite graph where each node represents the drug and target. The clear line edges between two nodes indicate the known interaction (positive instances), and the dotted line edges are shown as unknown interactions (negative instances).

The DTI network is represented by an $m \times n$ shape of the adjacency matrix Y:

$$y_{ij} = \begin{cases} 1, \textit{ Known Interactions} \\ 0, \textit{Unknown Interactions} \end{cases} \quad (1)$$

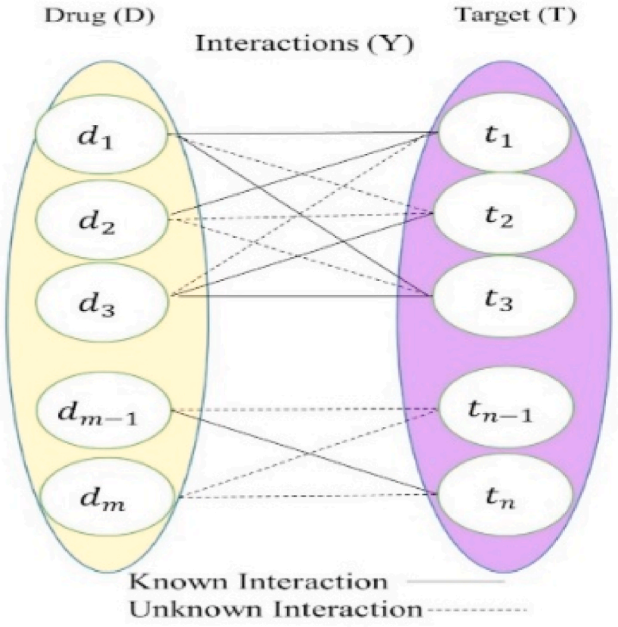

Fig. 2. A graphical view of DTIs.

where y_{ij}represents the ith and jth components of a matrix $(1 \le i \le m, 1 \le j \le n)$. The components with $y_{ij} = 0$and $y_{ij} = 1$correspond to negative (unknown) interactions and positive interactions, respectively.

3. Method

3.1. Dataset

The experimental dataset was freely collected from the Canadian online database (DrugBank) [5], which contains the drugs and targets related to information. Some information about the dataset used in this research is listed in Table 1. There are 5877 drugs and 3348 target proteins and a total of 12,674 interactions between those drugs and targets. Note that different versions of the DrugBank dataset have been exploited as benchmark datasets in previous research [3,34,35].

3.2. Drug-target feature extraction

A specific drug ID (e.g., D02361) is used to collect the molecular structure of drugs in SMILES format from the DrugBank database. Similarly, a target protein ID (e.g., has:1132) is used to obtain the FASTA sequence from the KEGG database. Most importantly, those drug IDs and protein IDs are both collected from the KEGG database before starting to discover the drug chemical structures and protein sequences for further processing. A variety of extraction techniques have been developed for generating features from drug chemical structures and protein sequences. Here, the drug chemical structure was calculated by the Rcpi [36] Toolkit, complete integrated cheminformatics, and bioinformatics package for drug discovery on multiple platforms (macOS, Linux, Windows). The extracted drug features include topological constitutional and geometrical features, which represent the complete numerical values of molecular properties. Note that the drugs with small molecules were removed in this study, as the Rcpi Toolkit could only process ideal molecular structures. The protein features were generated from their sequences using the PROFEAT [37] webserver. After applying the

Table 1
Statistics of the DrugBank dataset.

Drugs	Targets	Interactions
5877	3348	12,674

extraction techniques on the sequence, the protein features were encoded in amino acid composition (AAC), pseudoamino acid composition (PAAC), autocorrelation, dipeptide composition, quasi-sequence-order and CTD (Composition, Transition, Distribution). Before using these features in the classifiers, they are accurately normalized. The extraction process can be found on the PROFEAT webpage or online documents.

After generating the features for each of the drug structures and target sequences, they are represented into fixed-length vectors, which can be considered as input in the classifier. Some drug and target features were removed, which has no contribution to the prediction process. Finally, 193 and 1290 features were obtained for each drug (or target). Next, the drug and target are combined to form a drug-target pair by feature vectors. For example, the features of a pair (d, t) are represented by $[d_1, d_2 \ldots\ldots d_{193}, t_1, t_2, \ldots\ldots t_{1290}]$, where from a 193-dimensional feature vector for drug and 1290-dimensional feature vector for protein, a final 1483-dimensional feature vector is obtained for each pair. Here, drug-target pairs refer to instances. Using techniques to extract proteins holds not only adequate physicochemical information but also retains sequence-related information of target proteins.

3.3. Majority and minority instances balancing (MMIB)

We know that the drug-target dataset is highly imbalanced, where considerable instances are in the majority class, and few instances are in the minority class [2,39]. If the prediction model is trained with this imbalanced dataset, the classifier can lose the predictive ability to give accurate results. To overcome this situation, we have developed a data balancing technique called MMIB to manage the majority and minority instances in the datasets. If N is the total number of drug-target instances in the imbalanced dataset, it contains majority instances (MAi) and minority instances (MIi). Here, the size of MAi and MIi is represented by S_{MAi} and S_{MIi}. In these datasets, the amount of S_{MAi} is larger than S_{MIi}. We apply clusters in instances and make it into k clusters. If the MAi and MIi for the i th cluster $(1 \leq i \leq k)$ are S_{MAi} and S_{MIi}, respectively. The number ratio of MAi and MIi is S_{MAi}/S_{MIi}. In the training datasets, the ratio of S_{MAi} and S_{MIi} is set to $m:1$ $(m \geq 1)$. The selected MAi (i th cluster) is represented using the following expression [38]:

$$SS^i_{MAi} = (m \times S_{MIi}) \times \frac{S^i_{MAi} / S^i_{MIi}}{\sum_{i=1}^{k} S^i_{MIi} / S^i_{MIi}} \tag{2}$$

In equation (2), $m \times S_{MIi}$ is the majority of instances and $\sum_{i=1}^{k} S^i_{MIi} / S^i_{MIi}$ is the instance ratio of the majority and minority instances in the clusters. If the minority instances are absent in the cluster, then the S_{MIi} instances are considered as one; there should be a minimum of one minority instance in the cluster. We randomly select majority instances after determining the size of the majority instances using equation (2) in the cluster (). The total selected majority instances are $m \times S_{MIi}$ after merging them into each cluster. Finally, the selected majority instances and whole minority instances are combined to create a training dataset. In the training set, the ratio between S_{MAi} and S_{MIi} is $m:1$. The sequential steps of our balancing technique MMIB are shown in Table 2.

3.4. Convolutional neural network

Convolutional neural networks (CNNs or ConvNets) are a class of deep learning frameworks that are most frequently applied in various applications, such as recommender systems, video and image recognition, and natural language processing [40,41]. They are also known as space invariant ANN or shift-invariant because of its translation invariance nature and shared-weights structure. Generally, CNNs consist of an input and output layer and multiple hidden layers (MHLs). In CNNs, multiple hidden layers contain many convolutional layers (CLs). A regularization strategy called the dropout layer is assigned to mitigate the model overfitting problem that randomly adds noise to the HLs. The nodes indicated by 'dropped out' do not join in backpropagation and do not assist in the forward pass. Here, the RELU layer is a common activation function and is consequently monitored by extra convolutions such as fully connected layers (FCLs), pooling layers (PLs), and normalization layers (NLs) [31].

Table 2
The sequential steps of MMIB.

1. Set the ration of S_{MAi} to S_{MIi} (training dataset)
2. Apply cluster to instances into clusters
   2.1. Select K data points as the initial centroids
   2.2 Repeat
   for each data point
   calculate the distance from x to each centered
   Assign x to the closest centered
   end for
   2.3 Recompute the centered using the current cluster memberships
   2.4. Until criteria does not change
3. Determine the selected MAi using equation (2) and then randomly choose the MIi (each cluster)
4. Combine the selected MAi and whole MIi to generate the training sets.

In the full architecture of CNNs, convolutional layers are the main structure block. Deeper CLs are used to train more significant features using sliding kernels on the upper part of the earlier layers. Usually, the pooling operation is applied after each CL, which adds responses at various locations and combines robustness to small spatial variations. Thus, PLs help decrease the number of provided features and offer translation invariance by local nonlinear functions. Therefore, it controls the convergence and reduces the computation of NN. FCLs contain output neurons that are the same as ordinary neural networks that are identified in the last CNN layers. Each of the neurons is fully connected with all the nodes in the previous and the next layers [38].

The method showed an effective improvement in terms of computational complexity and program runtime after adding one more CL and max-PL with the process. The outputs of each convolutional layer can be calculated using the following equation:

$$y^l_k = f\left(\sum_m W^l_{m,k} y^{l-1}_m + b^l_k\right), \tag{3}$$

where l represents the layer index; m and k are the input and output feature maps, respectively. More, specifically, y^l_k indicates the input of the kth feature map of the l layer, and y^{l-1}_m indicates the output of the mth feature map of layer $l-1$. W and b are the weight tensor and bias term, respectively. The logistic regression algorithm is simply set in the output layer of this model, where y^l_k is an input and is calculated as follows:

$$\check{y} = f\left(W^l y^l + b^l\right) \tag{4}$$

where $\check{y}$ represents the predicted score of the model. W and b are the weight matrix and the bias vector, respectively. Here, DTIs is a binary classification problem; therefore, the output value is 2, represented by a positive and negative class. We minimized the cross-entropy loss by using a back-propagation and adaptive estimation technique [42].

$$loss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \check{y}_l + \left(1 - y_i\right)\log\left(1 - \check{y}_l\right)\right] \tag{5}$$

Batch normalization [43] and dropout [44] tricks were used to increase the efficiency of the model. In the training process, the dropout drops a few units in FCLs, whereas batch normalization helps to normalize the inputs into unit standard deviation and zero mean. Moreover, dropout was able to handle the overfitting problem, and batch normalization supports the model using sufficient learning ratios.

3.5. Feature reduction-LASSO

We assume that a 1483-dimensional feature vector of each pair (drug-target) may hold some redundant and noisy information, which may have a detrimental impact on the model during the prediction task. Therefore, the LASSO technique was used to manage (reduce) the dimensionality of the drug-target feature where unnecessary information was removed to provide discriminating information for each pair in the original data. The LASSO technique first proposed by Tibshirani [45], which is the main idea behind this technique, is to decrease the penalty function in the constraint. The dimensionality reduction (DR) of the LASSO method for a given dataset is as follows [39]:

$$X = [x_1, x_2, \ldots, x_n]^T \in R^{n \times d} \tag{6}$$

where $x_i$indicates the i th sample of the eigenvector. N represents the total number of samples in the given dataset, and d represents the feature dimension. Here, the corresponding response vectors of samples are represented by $Y = [y_1, y_2, \ldots, y_n] \in R^n$. This study is a binary classification problem where $y_i \in \{0, 1\}$ indicates the sample class label. The optimization of the LASSO method was as follows:

$$\min \left|\left| Y - X^T w \right|\right|_2^2 + \gamma \left|\left| w \right|\right|_1 \tag{7}$$

where w_1 is the regularization term of eigenvectors that accept the L_1 paradigm to produce a sparse solution for the feature space. The redundant and irrelevant features can be set to 0 (coefficients), and nonzero coefficients can be considered for classification. The regularization parameter γ 0 is responsible for handling the model complexity and data fitting. Using this LASSO, all the features were further generated into a low-dimensional vector. Therefore, the refined and deeper features were used as input in the CNN classifier and easily recognized by the model.

4. Experimental results

The proposed techniques were fully implemented using Python language (version 3.6) on Pytorch and the scikit-learn library. To speed up the computational process, we ran our model on a Windows system with 2.30 GHz Intel Xeon Gold 6140 processor and 128 GB RAM. In this experiment, we performed different types of tests to verify the effectiveness of classifiers, balancing, and selection techniques applied in the study. Finally, we composed a list of the newly predicted interactions of our methods.

4.1. Performance evaluation

Different validation methods have been used for measuring the model performance. Here, we used a 5-fold CV test to evaluate our model. In 5-fold CV, the main dataset is first separated into five approximately equal segments where four segments fit for training and one segment fit for the validation of the model. Then, the model is trained based on the training data by setting the parameters. After that, all the validation metrics are calculated using test data. Finally, the same procedure is repeated five times to achieve the performance results of the model and then the average of each validation metric is calculated. Some metrics for evaluating the performance in the proposed framework include accuracy $= (TP + TN)/(TP + FP + TN + FN)$, sensitivity $= TP/(TP + FN)$, specificity $= TN/(TN + FP)$, MCC $= ((TP \times TN) - (FP \times FN))/\sqrt{((TP + FN) \times (TN + FP) \times (TP + FP) \times (TP + FN))}$ and F1 $= 2TP/(2TP + FP + FN)$. Herein, P represents the positive instances and N represents the negative instances in the experimental dataset. TP and TN represent the correctly predicted instances (true positives and negatives), FP represents negative samples incorrectly predicted as positive samples (false positives), and FN represents positive samples incorrectly predicted as negative samples (false negatives). Moreover, we also used the AUC metric to measure the performance of our deep ACTION method. The AUROC curve is plotted by TPR (sensitivity) and FPR (1-specificity) with different thresholds adjusted.

4.2. Performance comparison with different classifiers

To check the effectiveness and robustness of the deep ACTION model, a comparison was made with different classifiers on the DrugBank dataset. Table 3 shows the performance of CNNs [46], Gaussian Naïve Bayesian (GBN) [47], XGBoost [48] and k-Nearest Neighbor (KNN) [49]. From the experimental results, we can see that each of the classifiers achieved optimal performance in our model with extracted features through parameter tuning. These four classifiers obtained better AUC values in our experiments. However, the CNN classifier shows the highest performance compared to the other three classifiers, especially in AUC, where the AUC values of 0.9836 obtained by the CNNs and the second highest AUC value of 0.9435 were archived by the GBN (also see Fig. 3). We observed that the AUC of our CNNs in the DrugBank dataset reaches 0.9836, which is 4.01%, 9.80%, and 10.13% higher than GBN, XGBoost, and KNN, respectively. Furthermore, F1 is also an important evaluation metric to analyze the balance within precision and sensitivity ratios. Our deepACTION model attains the top F1 value (0.9710) among other classifiers, which indicates that CNNs are a suitable algorithm to predict associations between drugs and targets. Moreover, CNNs provide satisfactory performance in detecting the relationship between DTIs.

Moreover, the CNN classifier also obtains the best AUPR value of 0.9123 on the DrugBank dataset. Our proposed framework achieved AUPRs of 0.7889, 0.8056, 0.8432, and 0.9323 using KNN, XGBoost, GBN, and CNNs, respectively. These results indicate that the CNN classifier outperforms other classifiers on DrugBank datasets.

4.3. Performance evaluation on DTIs

The experimental dataset was divided into three parts to evaluate the effectiveness of the proposed deepACTION model: training samples, test samples, and validation samples in ratios of 0.7, 0.2, and 0.1, respectively. The training samples are utilized for the pretraining model, the validation samples are utilized to optimize the method parameter, and the test samples are utilized to assess the method. We used a 5-fold CV test on the dataset to obtain reliable and stable results.

Table 4 lists the average prediction performance of the model. Performance comparison of different classifiers on validation and test samples used a 5-fold CV, as shown in Fig. 4. In the case of validation samples, the AUC, Acc, Sen, Pre, F1 and AUPR of the CNN model reach 0.9734, 0.9798, 0.9675, 0.9765, 0.9854 and 0.9156, respectively, while on test samples, the AUC, Acc, Sen, Pre, F1 and AUPR of the CNN model reach 0.9836, 0.9744, 0.9712, 0.9823 0.9710 and 0.9323, respectively.

For the GBN classifier, the AUC, Acc, Sen, Pre, F1 and AUPR of our model reach 0.9278, 0.9234, 0.8456, 0.8865, 0.8878 and 0.8166, respectively, in the validation samples. However, for the test samples, the AUC, Acc, Sen, Pre, F1 and AUPR are 0.9435, 0.9156, 0.8598, 0.8765, 0.8689 and 0.8432, respectively, which are 4.01%, 5.88%, 11.14%, 10.58%, 10.21% and 8.91% lower than those using the CNN classifier. Compared with the XGBoost classifier, our model increases by 9.8%, 12.66%, 11.45%, 23.56%, 17.23% and 12.67% for test samples and 9.36%, 12.52%, 11.30%, 21.11%, 17.78% and 12.78% for validation samples on AUC, Acc, Sen, Pre, F1 and AUPR. Similarly, for the test

Table 3
Performance comparison among different classifiers on the DrugBank dataset.

Classifiers	AUC	Acc	Sen	Pre	F1	AUPR
KNN	0.8823	0.8245	0.9087	0.7786	0.8234	0.7889
XGBoost	0.8856	0.8478	0.8567	0.7456	0.7987	0.8056
GBN	0.9435	0.9156	0.8598	0.8765	0.8689	0.8432
CNNs	0.9836	0.9744	0.9712	0.9823	0.9710	0.9323

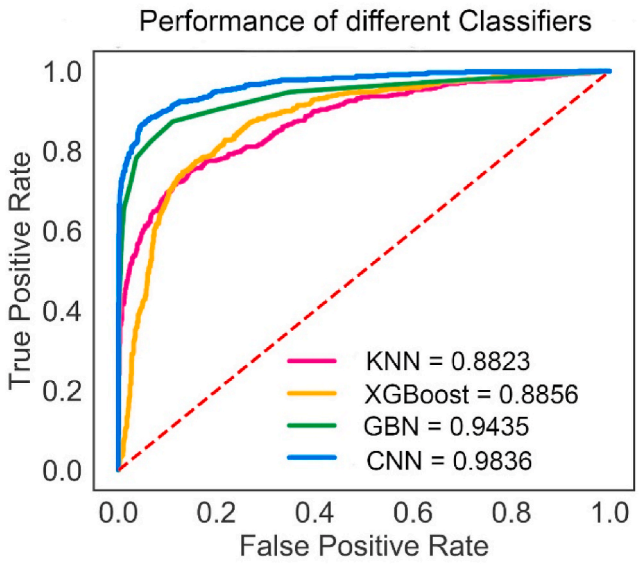

Fig. 3. ROC curves of different classifiers.

samples, the AUC, Acc, Sen, Pre, F1, and AUPR of the KNN classifier reach 0.8823, 0.8245, 0.9087, 0.7786, 0.8234, and 0.7889, respectively. It is 10.13%, 14.99%, 6.25%, 20.37%, 14.76% and 14.34% lower than the CNN classifier on AUC, Acc, Sen, Pre, F1 and AUPR. The results obtained by CNNs are much better than the other three classifiers on the DrugBank dataset; the average performance for metrics is more than 97%, significantly representing the benefit of feature extraction, balancing, and selection techniques of our proposed method. CNNs provides a deep learning-based model that is easy to use for drug-target applications. It gives more accurate prediction performances on huge features with a lower tendency for model overfitting.

CNNs achieved higher precision values than sensitivity on our experimental dataset, which indicates that our deepACTION model has a powerful ability to infer new interactions from negative samples. We already know that the new interactions are predicted from the negative drug-target samples. Here, our powerful CNN-based model incorporates drug features and protein features (extracted using different protein extraction techniques) for effective, accurate, and robust prediction of DTIs. Fig. 5 represents the results of the validation samples and test samples for the best two classifiers, CNNs and GBN. In Fig. 5, the ROC curve generated by our model is higher than other methods. It is clear that our approach can train extracted features to detect unknown DTIs.

4.4. Effect of the dimensionality reduction technique on the balance and imbalance dataset

Dimensionality reduction (DR) techniques can reduce the redundant features in the dataset, which can boost the prediction performance of the model. The 193-dimensional feature vectors generated from drug chemical structures and the 1290-dimensional feature vectors extracted from the target sequences, choosing a suitable technique to increase the performance rate, is of the most significance to the foundation of the predictive model. In this paper, we compared the impact of LASSO [45], PCA [50], ReliefF [51], and Elastic Net [52] on balanced and imbalanced datasets for selecting a suitable DR technique. Among them, the contribution rate is selected to be more than 86% when the DR is performed by PCA, and 30% features are considered for the ReliefF technique after arranging according to the weight size. Moreover, applying Elastic Net and LASSO for DR, the feature coefficient is selected to be nonzero, and the rest of the parameters are set to default. For a fair comparison between different DR techniques, in this experiment, the CNN classifier and 5-fold CV are used to identify and evaluate the model performance. The performance of different feature selection techniques

Table 4
Performance of different classifiers on validation samples and test samples.

Classifiers	Methods	AUC	Acc	Sen	Pre	F1	AUPR
KNN	Validation Samples	0.8803	0.8289	0.8934	0.7834	0.8301	0.7701
	Test Samples	0.8823	0.8245	0.9087	0.7786	0.8234	0.7889
XGBoost	Validation Samples	0.8798	0.8546	0.8545	0.7654	0.8076	0.7878
	Test Samples	0.8856	0.8478	0.8567	0.7456	0.7987	0.8056
GBN	Validation Samples	0.9278	0.9234	0.8456	0.8865	0.8878	0.8166
	Test Samples	0.9435	0.9156	0.8598	0.8765	0.8689	0.8432
CNNs	Validation Samples	0.9734	0.9798	0.9675	0.9765	0.9854	0.9156
	Test Samples	0.9836	0.9744	0.9712	0.9823	0.9710	0.9323

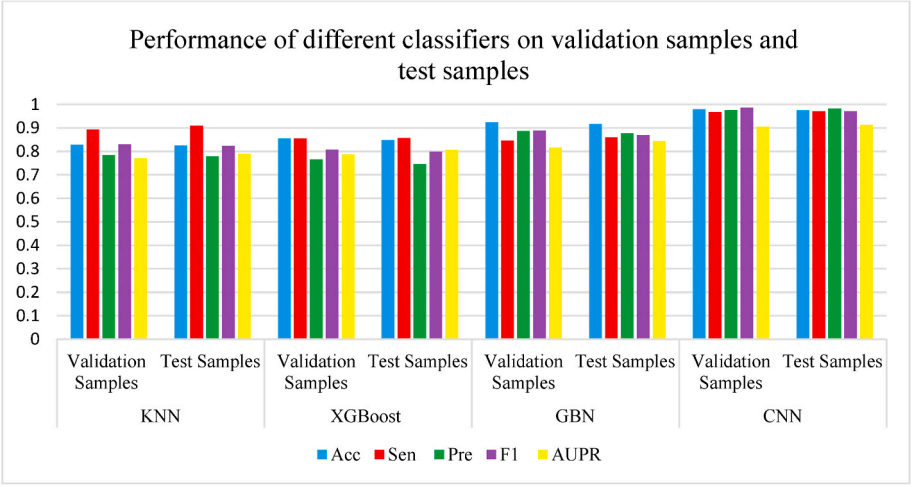

Fig. 4. Performance comparison of different classifiers on validation samples and test samples using 5-fold CV.

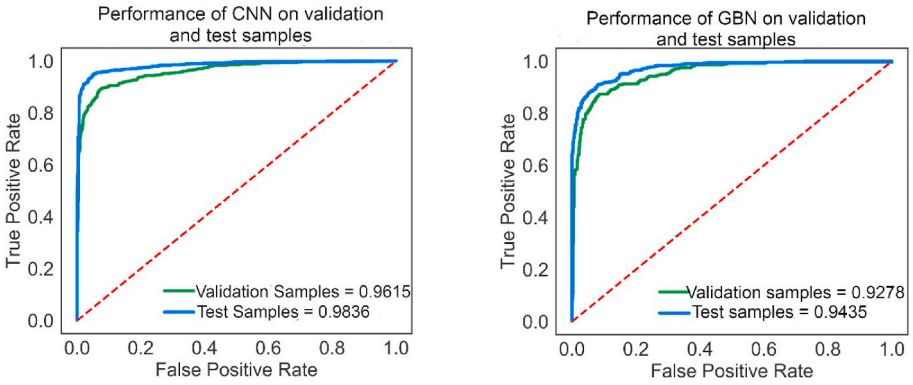

Fig. 5. ROC curves on validation and test samples.

on balanced and imbalanced data are listed in Table 5, and their results graph is shown in Fig. 6.

In Table 5 and Fig. 6, we can see that the LASSO and PCA techniques have a great influence on the DrugBank dataset. Furthermore, the LASSO technique showed the highest results in the four techniques. For an imbalanced dataset, using LASSO as the DR technique, AUC, Acc, Sen, Pre, F1 and AUPR reach 0.9616, 0.9584, 0.9452, 0.9515, 0.9710 and 0.9055, respectively. Similarly, for balanced data by MMIB, AUC, Acc, Sen, Pre, F1 and AUPR reach 0.9836, 0.9744, 0.9712, 0.9823, 0.9710 and 0.9323, respectively. The LASSO technique archived better results for a balanced dataset, which is 2.2% and 2.6% improvement in terms of AUC and AUPR metric. Moreover, using the LASSO technique for DR has great advantages over other techniques on all metrics. There was a serious imbalance problem in the drug-target dataset that can be responsible for reducing the prediction ability of the model. However, our developed MMIB technique helps to manage the imbalance problem and balance the interacting (minority) and noninteracting (majority) pairs to overcome the model complexity. To achieve the desired results and reduce the majority samples of the dataset on the CNN classifier, the MMIB is applied to balance the negative and negative pairs for the selected data by the LASSO technique. For a fair comparative analysis of the model, we conducted experiments on the reduced features that contain structural and sequence information of the drug-target data. The evaluation metric AUC has no significance in performing the merits of the classifier when the dataset is imbalanced. Most importantly, the negative samples are much greater than the negative samples in the imbalance dataset; therefore, other mentioned evaluation metrics have no accurate effect on the prediction results of the proposed model.

In Fig. 7, it can be clearly seen that the LASSO model has obtained a significant improvement in the matric AUC after using MMIB. After the DrugBank dataset is balanced, the metric AUC increases by 0.9836 to 0.9616. Moreover, the evaluation metric AUC increases by 1.78%, 2.10%, 2.30%, and 2.20% in the Elastic Net, ReliefF, PCA, and LASSO for balanced data, respectively. From the above discussion, we can say that the proposed model shows a great influence on the LASSO and MMIB techniques, which helps to handle the complexity and try to avoid overfitting of the model. Therefore, LASSO and MMIB are suitable feature selection and balancing techniques with the drug-target dataset in this study.

4.5. Hyperparameter adjustment

We discussed two primary hyperparameters, the batch normalization (BN) layer and the learning rate (LR) of our CNNs in this section. LR has a significant influence on CNNs. A larger LR runs the CNN gradient faster. It is difficult to converge CNNs with smaller LRs and requires a long execution time to train the full model. Here, we investigated five different ranges (1e-2∼1e-6) of LR for our model. If LR was set to 0.01, then CNNs achieved an Acc of 0.59, which confirmed that the model missed the important features during training to detect DTIs. Our model performed well with LR on the scale of 1e-3∼1e-6, where the Acc is more than 0.97 (see Table 6). Moreover, the performance of the model is increased with an LR of 1e-4.

BN is an active parameter to increase the efficiency of the classifier. The BN layer can normalize the input data and facilitate the training procedure. Therefore, the predictive model obtained the highest performance with the BN layer, which is 4% higher than that without BN. The performance results demonstrated that the BN layer could perfectly normalize input data in a reverse range; therefore, the proposed model can effectively train input features to predict DTIs.

4.6. Identifying new DTIs

In this section, we investigate the capability of deepACTION in predicting novel interactions. Here, DrugBank databases were considered to ensure whether novel identified interactions are present. The interactions (DTIs), which are detected by deepACTION, were ranked based on their interaction (drug-target pair) probabilities, as listed in Tables 7 and 8. Generally, the drug and protein-related databases are kept updated; the new interactions are stored in the web portal, such as ChEMBL, KEGG, and DrugBank. After examining the predicted pairs by our method, the highly scored interactions not only obtained good AUC but also pharmacologically realistic new interactions. For simplicity, we considered high scored pairs as new interactions for the first top-rated pairs, and if a new pair is presented in the last version of the DrugBank database, then we can say the interaction is validated.

In particular, we conducted an investigation on one drug, cabergoline (DB00248), and one target, estrogen receptor (P03372). Tables 7 and 8 report the top predicted values for the cabergoline and estrogen

Table 5
Comparison of different feature selection techniques on imbalanced and balanced data.

Methods	Imbalanced Data without MMIB						Balanced Data with MMIB					
	AUC	ACC	Sen	Pre	F1	AUPR	AUC	ACC	Sen	Pre	F1	AUPR
Elastic Net	0.8578	0.8478	0.8212	0.8345	0.8435	0.8045	0.8756	0.8645	0.8458	0.8523	0.8675	0.8356
ReliefF	0.8756	0.8520	0.8538	0.8508	0.8245	0.7923	0.8967	0.8823	0.8756	0.8710	0.8612	0.8267
PCA	0.9315	0.9156	0.9167	0.9115	0.9278	0.8814	0.9545	0.9478	0.9369	0.9434	0.9431	0.9134
LASSO	**0.9616**	**0.9584**	**0.9452**	**0.9514**	**0.9710**	0.9055	**0.9836**	**0.9744**	**0.9712**	**0.9823**	**0.9710**	0.9323

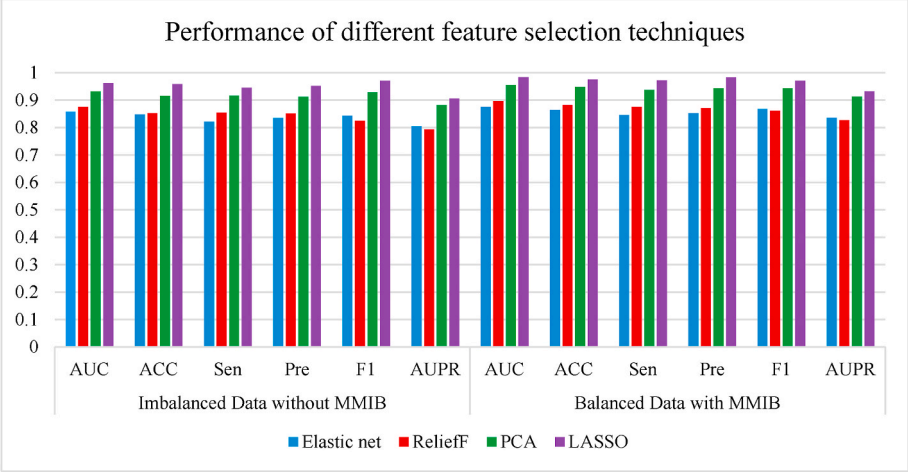

Fig. 6. Performance comparison of different feature selection techniques on balanced and imbalanced data.

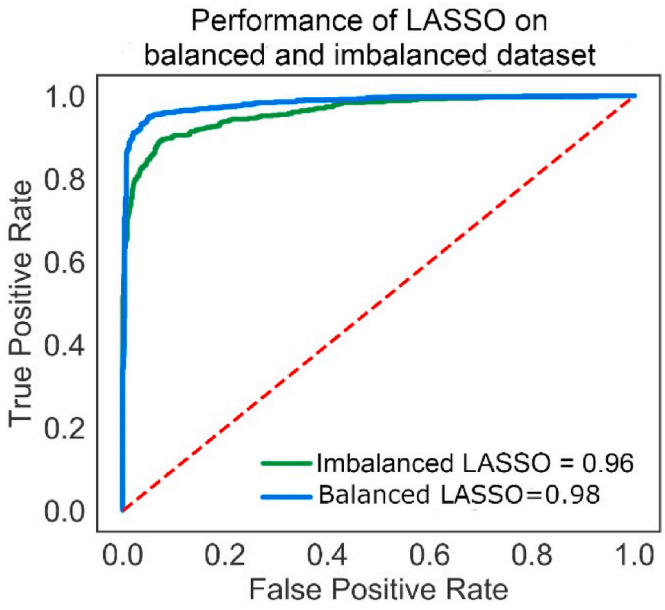

Fig. 7. ROC curve for balanced and imbalanced LASSO techniques.

Table 6
Performance of the proposed method with a different LR.

Parameters	AUC	Acc	Sen	Pre	F1	AUPR
LR = 1e-2	0.9323	0.5945	0.8478	0.8076	0.8156	0.8954
LR = 1e-3	0.9712	0.9546	0.9534	0.9656	0.9539	0.9112
LR = 1e-4	0.9836	0.9744	0.9712	0.9823	0.9712	0.9323
LR = 1e-5	0.9787	0.9612	0.9456	0.9623	0.9543	0.9178
LR = 1e-6	0.9167	0.9001	0.9108	0.9104	0.9121	0.8567

receptors. In our experiments, cabergoline interacted with a total of 21 targets. From the top 20 identified targets for cabergoline, 15 were accurately identified, as listed in Table 7. Moreover, the estrogen receptor has a total of 41 predicted drugs from the DrugBank dataset with our method. From the top 20 predicted drugs for the estrogen receptor, 15 were effectively identified, as listed in Table 8. These predictions (drugs and targets) show that our deepACTION model is truly effective in identifying novel interactions for any new dataset. The unconfirmed predictions might be correct after investigating their possibility. We can see from Fig. 8 that a total of 30 interactions are predicted successfully

Table 7
Top 15 targets predicted for Cabergoline.

Drug Name: Cabergoline		
Rank	Target ID	Target Name
1	P35462	D (3) dopamine receptor
2	P28223	5-hydroxytryptamine receptor 2A
3	P18089	Alpha-2B adrenergic receptor
4	P28221	5-hydroxytryptamine receptor 1D
5	P14416	D (2) dopamine receptor
6	P41595	5-hydroxytryptamine receptor 2B
7	P08908	5-hydroxytryptamine receptor 1A
8	P18825	Alpha-2C adrenergic receptor
9	P21917	D (4) dopamine receptor
10	P08913	Alpha-2A adrenergic receptor
11	P28222	5-hydroxytryptamine receptor 1B
12	P28335	5-hydroxytryptamine receptor 2C
13	P34969	5-hydroxytryptamine receptor 7
14	P21918	D (1B) dopamine receptor
15	P21728	D (1A) dopamine receptor

Table 8
Top 15 drugs predicted for estrogen receptor.

Target Name: Estrogen receptor		
Rank	Drug ID	Target Names
1	DB00255	Diethylstilbestrol
2	DB00304	Desogestrel
3	DB00269	Chlorotrianisene
4	DB00294	Etonogestrel
5	DB00286	Conjugated estrogens
6	DB00367	Levonorgestrel
7	DB00396	Progesterone
8	DB00539	Toremifene
9	DB00431	Lindane
10	DB00396	Progesterone
11	DB00481	Raloxifene
12	DB00603	Medroxyprogesterone acetate
13	DB00539	Toremifene
14	DB00624	Testosterone
15	DB00655	Estrone

from our experimental data for drug-cabergoline, a protein-estrogen receptor using the deepACTION method. These experimental results disclose that our deepACTION approach could be utilized to predict the new DTI network, indicating that the proposed techniques are more practical in real applications.

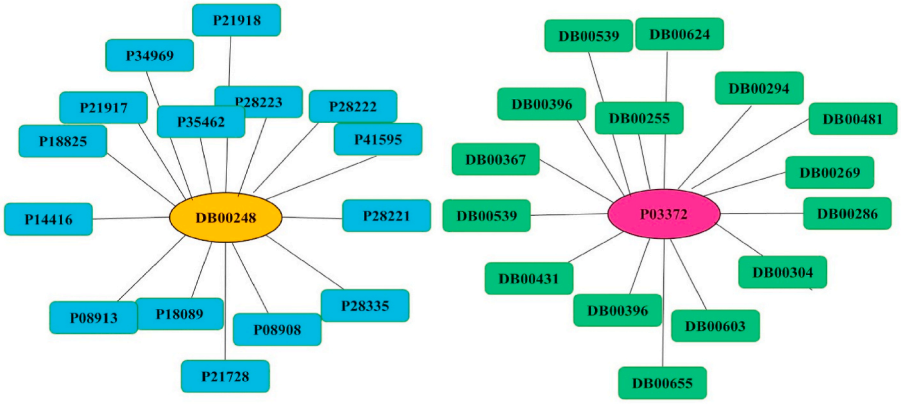

Fig. 8. Results of novel predicted DTIs.

4.7. Verify the proposed method using gold standard datasets

Although the proposed deepACTION model has attained satisfactory performance with the DrugBank dataset in this study, it is necessary to confirm whether our developed model can achieve better performances on gold standard datasets. Therefore, the experiments validate the proposed model using four datasets, namely, Nuclear Receptors (NR), GPCRs (G), Ion Channels (I), and Enzymes (E). The positive interaction pairs between these drugs and targets are 90, 635, 1476, and 2926, respectively, and finally, a total of 5127 known interactions were obtained from these datasets.

We use similar techniques for extracting drug and target features from these gold standard datasets and then apply the LASSO technique to reduce the features as before. Moreover, our MMIB algorithm uses the datasets to create negative samples for effective prediction. Finally, the processed data feeds to KNN, XGBoost, GBN, and CNN classifiers. The performance AUC results for the four classifiers using these external datasets are listed in Table 9. The results in Table 9 show that the CNN classifier obtained the best performance AUC of 0.0.9786 for the enzyme (E) dataset. In addition, the second-highest result is achieved by the GBN classifier, where the performance AUC value is 0.9577. The AUC values obtained by the CNN classifier for the four drug-target datasets are comparatively higher than those obtained by the GBN, XGBoost, and KNN classifiers. By comparing the AUC values, we can observe that the CNN classifier is 2.09%, 10.47%, and 10.25% higher than the other classifiers. For the ion channel (I) dataset, again, the top results obtained by the CNN classifier and second-highest result are also achieved by GBN. More specifically, the CNN classifier performs very well for the enzyme (E) and ion channels (I). This means that CNNs consistently attained effective prediction performances. The KNN and XGBoost prediction performances are relatively similar in some cases, where the AUC values are 0.8761 and 0.8739 for the enzyme (E) dataset.

Table 9
Performance of different classifiers on gold standard datasets.

Datasets	Classifiers	AUC
Enzymes (E)	KNN	0.8761
	XGBoost	0.8739
	GBN	0.9577
	CNNs	0.9786
Ion Channels (I)	KNN	0.8377
	XGBoost	0.8676
	GBN	0.9097
	CNNs	0.9687
GPCRs (G)	KNN	0.8428
	XGBoost	0.8468
	GBN	0.9088
	CNNs	0.9655
Nuclear Receptors (NR)	KNN	0.8724
	XGBoost	0.8749
	GBN	0.9588
	CNNs	0.9620

We can conclude that these three classifiers are not suitable to identify DTIs effectively. Therefore, CNNs are considered classifiers for our method that might be able to predict DTIs on huge datasets. The AUC values for different classifiers from four datasets are depicted in Fig. 9, where the graph produced by CNNs is significantly higher than the other three classifiers.

4.8. Comparison with other methods and verifying the proposed method using gold standard datasets

In 5-fold CV, we compared the proposed method deepACTION with the other three previous methods, including BE-DTI [35], Ezzat et al. [34] and Yu et al. [14]. The results of the performance comparison on the DrugBank dataset in terms of AUC are listed in Table 10, which shows that the AUC values of our deepACTION model are higher than those of the other three approaches. For instance, compared with the BE-DTI approach, which integrated different dimensionality reduction techniques and active learning to manage the high dimension features and imbalance problem of the DrugBank dataset, the AUC of deepACTION is 0.983 greater than that of BE-DTI. Our method attains AUC improvements of 5.6% compared to BE-DTI. Compared with Ezzat et al. [34], who proposed an ensemble learning framework to solve the imbalance problem (between-class and within-class) in the same dataset, generating the drug-target features in a low-dimension vector with various extraction techniques and predicting interactions from existing drugs and targets, the AUC of deepACTION is greater than that of Ezzat et al. [34]. As shown in Table 10, deepACTION also achieves the best result compared to the Yu et al. [14] methods. Finally, we observed that our method achieved improved AUC results compared to the three other methods.

Next, we compared the deepACTION method to seven methods from the literature, those of Huang et al. [53], Mousavian et al. [54], Li et al. [55], Rayhan et al. [56], Gönen [16] and Yamanishi et al. [57]. All these methods used four gold-standard datasets as experimental datasets. Table 11 shows the average AUC results for those methods, including deepACTION. From the boldfaced fonts in Table 11, we observe that the model deepACTION significantly outperformed the previous methods for all four datasets in terms of the AUC metric. Our method is generally more suitable for any size dataset, regardless of whether the dataset has a large number of samples with huge features or a small number of features. The average values attained for our method on the datasets of enzyme, ion channel, GPCR, and nuclear receptor are 0.9786, 0.9687, 0.9655, and 0.9620, respectively. We can see from Table 11 that the deepACTION model provides consistent performers for all the datasets.

We found some reasons for the impressive performances of deepACTION. DeepACTION takes the benefit of the CNN model that can

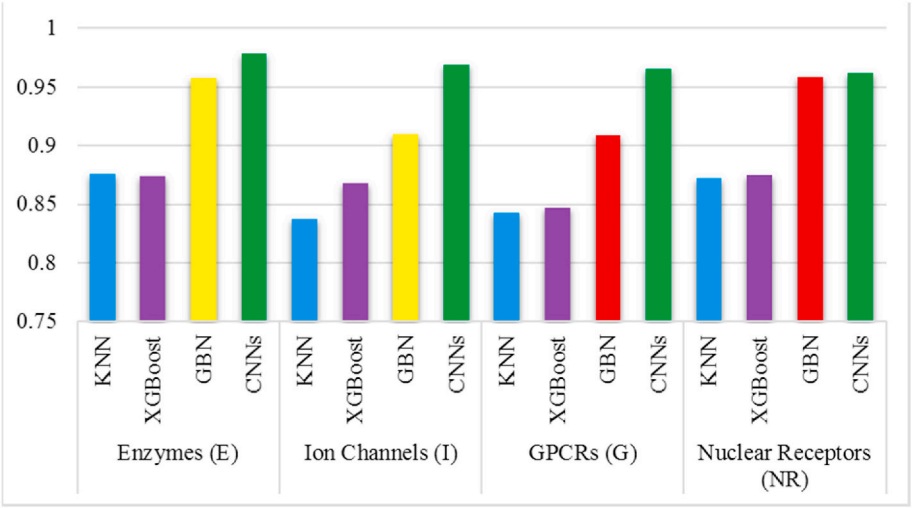

Fig. 9. Performance comparison of different classifiers on gold standard datasets using 5-fold CV. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 10
Comparison of deepACTION with existing methods on the DrugBank dataset.

Methods	AUC
BE-DTI [35]	0.927
Ezzat et al. [34]	0.900
Yu et al. [14]	0.905
deepACTION	**0.983**

Table 11
Performance comparison of deepACTION with existing methods on the gold standard dataset.

Methods	Enzyme	Ion Channel	GPCR	Nuclear Receptor
Huang et al. [53]	0.9040	0.8510	0.8990	0.8430
Mousavian et al. [54]	0.9480	0.8890	0.8720	0.8690
Li et al. [55]	0.9288	0.9171	0.8856	0.9300
Rayhan et al. [56]	0.9754	0.9512	0.9478	0.9241
Gönen [16]	0.8320	0.7990	0.8570	0.8240
Yamanishi et al. [57]	0.8075	0.8029	0.8022	0.7578
deepACTION	**0.9786**	**0.9687**	**0.9655**	**0.9620**

quickly learn large attractive and valid features to reach credible and satisfactory performances and introduces the combined technique LASSO-CNN to manage the large-scale dimensions of data with low computational complexity.

5. Conclusion

In this study, a novel deep learning-based model, deepACTION, was proposed using the CNN algorithm to determine the interactions between drugs and targets. Different feature extraction techniques were utilized to represent the drug-target structure (chemical structure and protein sequence) in numerical form. Most importantly, most of the related work used random sampling to manage the imbalanced datasets, but we newly developed a balancing technique, MMIB, to handle the majority (negative) and minority (negative) instances in the datasets. MMIB offers a powerful mechanism to enhance model performance. Moreover, the LASSO technique was applied to convert the higher dimensional (drug-target features) feature space into a lower-dimensional feature space and make the training process easier. We compared our method with existing algorithms under a validation test (5-fold CV), and the experimental results show that our method achieved prediction performance in terms of all measurement metrics and was able to predict novel pairs (drug-target) from the DrugBank dataset. The improved performance of deepACTION may inspire scientists to use this method in predicting new DTIs. In the future, we shall consider a heterogeneous drug-target (networks) dataset with auPR matrices for experiments and develop a separate web application by providing simple mechanisms and a user-friendly interface for our model.

Author Contributions

S M Hasan Mahmud: conceived and designed the experiments and performed the experiments, analyzed the data and contributed analysis tools, Writing - original draft, wrote the paper. Wenyu Chen: analyzed the data and contributed analysis tools, Writing - original draft, wrote the paper. Hosney Jahan: Writing - original draft, wrote the paper. Bo Dai: Writing - original draft, wrote the paper. Salah Ud Din: Writing - original draft, wrote the paper. Anthony Mackitz Dzisoo: Writing - original draft, wrote the paper

Acknowledgments

This work was supported by the National Natural Science Foundation of China-Research on New Technology of Core Algorithm under Grant 61772115.

References

[1] S.M.H. Mahmud, W. Chen, H. Meng, H. Jahan, Y. Liu, S.M.M. Hasan, Prediction of drug-target interaction based on protein features using undersampling and feature selection techniques with boosting, Anal. Biochem. 589 (2020), https://doi.org/10.1016/j.ab.2019.113507.
[2] S.M.H. Mahmud, W. Chen, H. Jahan, N.I. Sujan, S. Ahmed, iDTi-CSsmoteB : identification of drug – target interaction based on drug chemical structure and protein sequence using XGBoost with over-sampling technique SMOTE, IEEE Access 7 (2019) 48699–48714, https://doi.org/10.1109/ACCESS.2019.2910277.
[3] J. You, R.D. Mcleod, P. Hu, Predicting drug-target interaction network using deep learning model, Comput. Biol. Chem. 80 (2019) 90–101, https://doi.org/10.1016/j.compbiolchem.2019.03.016.
[4] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, M. Tanabe, KEGG for integration and interpretation of large-scale molecular data sets, Nucleic Acids Res. 40 (2012) D109–D114, https://doi.org/10.1093/nar/gkr988.
[5] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A.C. Guo, D.S. Wishart, DrugBank 3.0: a comprehensive resource for "Omics" research on drugs, Nucleic Acids Res. 39 (2011) D1035–D1041, https://doi.org/10.1093/nar/gkq1126.
[6] A.P. Bento, A. Gaulton, A. Hersey, L.J. Bellis, J. Chambers, M. Davies, F.A. Krüger, Y. Light, L. Mak, S. McGlinchey, J.P. Nowotka, M. Papadatos, G. Santos,

R. Overington, The ChEMBL bioactivity database: an update, Nucleic Acids Res. 42 (2013) D1083–D1090, https://doi.org/10.1093/nar/gkt1031.
[7] D. Szklarczyk, A. Santos, C. Von Mering, L.J. Jensen, P. Bork, M. Kuhn, Stitch 5: augmenting protein-chemical interaction networks with tissue and affinity data, Nucleic Acids Res. 44 (2016) D380–D384, https://doi.org/10.1093/nar/gkv1277.
[8] F. Zhu, B. Han, P. Kumar, X. Liu, X. Ma, X. Wei, L. Huang, Y. Guo, L. Han, C. Zheng, Y. Chen, Update of TTD: therapeutic target database, Nucleic Acids Res. 38 (2010).
[9] J.B.O. Mitchell, The relationship between the sequence identities of alpha helical proteins in the PDB and the molecular similarities of their ligands, J. Chem. Inf. Comput. Sci. 41 (2001) 1617–1622, https://doi.org/10.1021/ci010364q.
[10] M.J. Keiser, B.L. Roth, B.N. Armbruster, P. Ernsberger, J.J. Irwin, B.K. Shoichet, Relating protein pharmacology by ligand chemistry, Nat. Biotechnol. 25 (2007) 197–206, https://doi.org/10.1038/nbt1284.
[11] M. Campillos, M. Kuhn, A.C. Gavin, L.J. Jensen, P. Bork, Drug target identification using side-effect similarity, Science 321 (2008) 263–266, https://doi.org/10.1126/science.1158140, 80-.
[12] Z. Mousavian, A. Masoudi-Nejad, Drug–target interaction prediction via chemogenomic space: learning-based methods, Expet Opin. Drug Metabol. Toxicol. 10 (2014) 1273–1287, https://doi.org/10.1517/17425255.2014.950222.
[13] A.S. Rifaioglu, H. Atas, M.J. Martin, R. Cetin-Atalay, V. Atalay, T. Doğan, Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases, Briefings Bioinf. (2018) 1–35, https://doi.org/10.1093/bib/bby061.
[14] H. Yu, J. Chen, X. Xu, Y. Li, H. Zhao, Y. Fang, X. Li, W. Zhou, W. Wang, Y. Wang, A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data, PLoS One 7 (2012), https://doi.org/10.1371/journal.pone.0037608.
[15] Z. He, J. Zhang, X.H. Shi, L. Le Hu, X. Kong, Y.D. Cai, K.C. Chou, Predicting drug-target interaction networks based on functional groups and biological features, PLoS One 5 (2010), https://doi.org/10.1371/journal.pone.0009603.
[16] M. Gönen, Predicting drug-target interactions from chemical and genomic kernels using Bayesian matrix factorization, Bioinformatics 28 (2012) 2304–2310, https://doi.org/10.1093/bioinformatics/bts360.
[17] K. Bleakley, Y. Yamanishi, Supervised prediction of drug-target interactions using bipartite local models, Bioinformatics 25 (2009) 2397–2403, https://doi.org/10.1093/bioinformatics/btp433.
[18] J. Mei, C. Kwoh, P. Yang, X. Li, J. Zheng, Drug-Target Interaction Prediction by Learning From Local Information and Neighbors 29 (2013) 238–245.
[19] X. Chen, M.X. Liu, G.Y. Yan, Drug-target interaction prediction by random walk on the heterogeneous network, Mol. Biosyst. 8 (2012) 1970–1978, https://doi.org/10.1039/c2mb00002d.
[20] Y.-C. Wang, Z.-X. Yang, Y. Wang, N.-Y. Deng, Computationally probing drug-protein interactions via support vector machine, Lett. Drug Des. Discov. 7 (2010) 370–378, https://doi.org/10.2174/157018010791163433.
[21] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, M. Kanehisa, Prediction of drug-target interaction networks from the integration of chemical and genomic spaces, Bioinformatics 24 (2008) 232–240, https://doi.org/10.1093/bioinformatics/btn162.
[22] Y. Huang, Z. You, X. Chen, A Systematic Prediction of Drug-Target Interactions Using Molecular Fingerprints and Protein Sequences, 2016, https://doi.org/10.2174/1389203718666161122103057.
[23] P.W. Hu, K.C.C. Chan, Z.H. You, Large-scale prediction of drug-target interactions from deep representations, in: Int. Jt. Conf. Neural Networks, 2016, pp. 1236–1243, https://doi.org/10.1109/IJCNN.2016.7727339.
[24] M. Hamanaka, K. Taneishi, H. Iwata, J. Ye, J. Pei, J. Hou, CGBVS-DNN : Prediction of Compound-Protein Interactions Based on Deep Learning, (n.d.) 1–11. doi: 10.1002/minf.201600045..
[25] K. Tian, M. Shao, Y. Wang, J. Guan, S. Zhou, Boosting compound-protein interaction prediction by deep learning, Methods (2016), https://doi.org/10.1016/j.ymeth.2016.06.024.
[26] M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, H. Lu, Deep learning-based drug-target interaction prediction, J. Proteome Res. 16 (2017) 1401–1409, https://doi.org/10.1021/acs.jproteome.6b00618.
[27] L.E.I. Wang, Z. You, X. Chen, W.E.T. Al, A Computational-Based Method for Predicting Drug – Target Interactions by Using Stacked Autoencoder Deep Neural Network, 24, 2017, pp. 1–13, https://doi.org/10.1089/cmb.2017.0135.
[28] T. Pahikkala, A. Airola, S. Pietila, Toward more realistic drug- target interaction predictions, Briefings Bioinf. (2014) 1–13, https://doi.org/10.1093/bib/bbu010.
[29] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, M. Ester, SimBoost : a read - across approach for predicting drug – target binding affinities using gradient boosting machines, J. Cheminf. (2017) 1–14, https://doi.org/10.1186/s13321-017-0209-z.
[30] E. Ozkirimli, DeepDTA : Deep Drug – Target Binding Affinity Prediction, 2018, https://doi.org/10.1093/bioinformatics/bty593.
[31] S. Hu, D. Xia, B. Su, P. Chen, B. Wang, J. Li, A convolutional neural network system to discriminate drug-target interactions, IEEE ACM Trans. Comput. Biol. Bioinf (2019), https://doi.org/10.1109/TCBB.2019.2940187.
[32] Y. Wang, Z. You, S. Yang, H. Yi, Z. Chen, K. Zheng, A deep learning-based method for drug- target interaction prediction based on long short-term memory neural network, BMC Med. Inf. Decis. Making 20 (2020) 1–9, https://doi.org/10.1186/s12911-020-1052-0.
[33] S. Hu, C. Zhang, P. Chen, P. Gu, J. Zhang, B. Wang, Predicting drug-target interactions from drug structure and protein sequence using novel convolutional neural networks, BMC Bioinf. 20 (2019) 1–12, https://doi.org/10.1186/s12859-019-3263-x.
[34] A. Ezzat, M. Wu, X.L. Li, C.K. Kwoh, Drug-target interaction prediction via class imbalance-aware ensemble learning, BMC Bioinf. 17 (2016), https://doi.org/10.1186/s12859-016-1377-y.
[35] A. Sharma, R. Rani, BE-DTI ': ensemble framework for drug target interaction prediction using dimensionality reduction and active learning, Comput. Methods Progr. Biomed. 165 (2018) 151–162, https://doi.org/10.1016/j.cmpb.2018.08.011.
[36] D. Cao, N. Xiao, Q. Xu, A. Chen, Rcpi : R/Bioconductor package to generate various descriptors of proteins , compounds and their interactions, Syst. Biol. (Stevenage). 31 (2015) 279–281, https://doi.org/10.1093/bioinformatics/btu624.
[37] H.B. Rao, F. Zhu, G.B. Yang, Z.R. Li, Y.Z. Chen, Update of PROFEAT : a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence 39 (2011) 385–390, https://doi.org/10.1093/nar/gkr284.
[38] S.J. Yen, Y.S. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Syst. Appl. 36 (2009) 5718–5727, https://doi.org/10.1016/j.eswa.2008.06.108.
[39] H. Shi, S. Liu, J. Chen, X. Li, Q. Ma, B. Yu, Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure, Genomics (2018) 1–14, https://doi.org/10.1016/j.ygeno.2018.12.007.
[40] Y. Lecun, Y. Bengio, G. Hinton, Deep Learning, 2015, https://doi.org/10.1038/nature14539.
[41] G.L. Grinblat, L.C. Uzal, M.G. Larese, P.M. Granitto, Deep learning for plant identification using vein morphological patterns, Comput. Electron. Agric. 127 (2016) 418–424, https://doi.org/10.1016/j.compag.2016.07.003.
[42] X. Glorot, A. Bordes, Deep sparse rectifier neural networks, in: 14th Int. Con-Ference Artif, Intell. Stat., 2011, pp. 315–323.
[43] C. Szegedy, S.G. Com, Batch Normalization : accelerating deep network training by reducing internal covariate shift, in: IProceedings 32nd Int. Conf. Int. Conf. Mach. Learn, 37, 2015, pp. 448–456.
[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout : a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[45] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc. 58 (2013) 267–288.
[46] L.D. LeCun, Yann, Boser, E. Bernhard, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, Jackel, handwritten digit recognition with a back-propagation network, in: Adv. Neural Inf. Process. Syst, 2, 1990.
[47] W.S. Geisler, R.L. Diehl, A Bayesian approach to the evolution of perceptual and cognitive systems, Cognit. Sci. 27 (2003) 379–402, https://doi.org/10.1016/S0364-0213(03)00009-0.
[48] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: 22nd ACM SIGKDD Int. Conf. Knowl. Discov, Data Min., 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.
[49] N.S. Altman, An introduction to kernel and nearest neighbor nonparametric regression, Am. Statistician 46 (1991) 175–185, https://doi.org/10.1080/00031305.1992.10475879.
[50] K.P.F.R. S, On lines and planes of closest fit to systems of points in space, Philos. Mag. 2 (2010) 559–572, https://doi.org/10.1080/14786440109462720.
[51] I. Robnik-Šikonja, M. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (2003) 23–69.
[52] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B. 67 (2005) 301–320.
[53] Y. Huang, Z. You, X. Chen, A systematic prediction of drug-target interactions using molecular fingerprints and protein sequences, Curr. Protein Pept. Sci. 19 (2018) 468–478, https://doi.org/10.2174/1389203718666161122103057.
[54] Z. Mousavian, S. Khakabimamaghani, K. Kavousi, A. Masoudi-Nejad, Drug-target interaction prediction from PSSM based evolutionary information, J. Pharmacol. Toxicol. Methods 78 (2016) 42–51, https://doi.org/10.1016/j.vascn.2015.11.002.
[55] Z. Li, P. Han, Z.H. You, X. Li, Y. Zhang, H. Yu, R. Nie, X. Chen, In silico prediction of drug-target interaction networks based on drug chemical structure and protein sequences, Sci. Rep. 7 (2017) 1–13, https://doi.org/10.1038/s41598-017-10724-0.
[56] F. Rayhan, S. Ahmed, Z. Mousavian, D. Farid, S. Shatabda, FRnet-DTI : deep convolutional neural network for drug-target interaction prediction, Heliyon 6 (2020), e03444, https://doi.org/10.1016/j.heliyon.2020.e03444.
[57] Y. Yamanishi, M. Kotera, M. Kanehisa, S. Goto, Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework, Bioinformatics 26 (2010) 246–254, https://doi.org/10.1093/bioinformatics/btq176.