Journal of Smart Algorithms and Applications
JSAA

Journal Homepage
https://pub.scientificirg.com/index.php/JSAA

Reliable Drug–Target Interaction Prediction Using Convolutional Neural Networks with Robust Negative Sample Generation

Heba Saeed[a], Mayar Adel[a], Yasmeen Ataa[a], Mayar Mohamed[a], Hasnaa Ahmed[b], Tan Wei Hong [c], Niresh Jayarajan [d,1]

[a] Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef City, 62511, Egypt Emails: Hebasaid214@gmail.com, Mayarelsisy10@gmail.com, yasmeenataa765_sd@fcis.bsu.edu.eg, mayar.shora2001@gmail.com
[b] Faculty of Computers and Artificial Intelligence, Fayoum University, Fayoum City, 63511, Egypt Email: ha2872@fayoum.edu.eg
[c] Faculty of Mechanical Engineering & Technology, University Malaysia Perlis (UniMAP), Main Campus Pauh Putra,02600 Arau, Perlis, Email: whtan@unimap.edu.my
[d,1] Department of Automobile Engineering, PSG College of Technology, INDIA, Email: jnr.auto@psgtech.ac.in

ABSTRACT - Proteins, including receptors, enzymes, and ion channels, represent primary biological targets whose interactions with small-molecule drugs play a critical role in therapeutic discovery and development. Accurate identification of drug–target interactions (DTIs) remains a fundamental challenge in drug discovery due to the high cost, time requirements, and scalability limitations of experimental validation. Consequently, computational approaches have emerged as efficient alternatives for large-scale DTI prediction. This study proposes a convolutional neural network (CNN)–based framework for predicting drug–target interactions, with a particular focus on reliable negative sample generation to address the inherent data imbalance and uncertainty present in DTI datasets. The proposed method incorporates feature projection techniques to effectively capture meaningful representations of drug and protein features while reducing noise and redundancy. By constructing more reliable negative instances, the framework improves model robustness and mitigates bias commonly introduced by randomly generated negative samples. The proposed model is evaluated on a benchmark DTI dataset, where it achieves a classification accuracy of 0.9800, demonstrating strong predictive capability. To further assess generalization performance, the model is tested on an independent external dataset derived from DrugBank. On this dataset, the framework attains an accuracy of 0.8814 and an Area Under the Receiver Operating Characteristic Curve (AUC) of 0.9527, indicating effective transferability across datasets. Experimental results confirm that the integration of CNN-based feature learning with reliable negative instance generation significantly enhances DTI prediction performance. The proposed framework offers a robust and generalizable computational tool for drug–target interaction prediction and has the potential to support early-stage drug discovery by reducing experimental search space and accelerating candidate prioritization.

PAPER INFORMATION

HISTORY
Received: 22 August 2025
Revised: 17 November 2025
Accepted: 19 January 2026
Online: 8 February 2026
MSC
62K05
62K15
KEYWORDS
Drug–target interaction (DTI) prediction, Convolutional neural networks (CNNs), Area under the ROC curve (AUC), Negative instances.

[d,1] Department of Automobile Engineering, PSG College of Technology, INDIA, Email: jnr.auto@psgtech.ac.in

1. INTRODUCTION

Developing a new pharmaceutical compound is both time-consuming and costly, often requiring approximately a decade for FDA approval [1]. Drug repurposing, also known as repositioning, seeks to identify new therapeutic uses for existing or previously discontinued drugs [2], [3]. Traditionally, this process has been limited to exploring a small set of drugs that engage with multiple targets across various diseases [4]. There is now growing recognition that a single compound may interact with numerous biological targets [5], [6], and could also affect proteins outside its primary therapeutic scope [7]. Consequently, mapping potential interactions between small molecules and proteins has become a vital component of drug discovery, helping reduce R&D costs and shedding light on the mechanisms underlying drug actions [8]. To address this, a wide range of silico methods have been proposed to predict drug-target interactions (DTIs). These approaches generally fall into two categories: ligand-based methods and molecular docking. Ligand-based techniques, such as Quantitative Structure–Activity Relationship (QSAR), use machine learning to infer interactions by comparing candidate ligands to those already known for the target. However, their performance often depends heavily on the availability of known ligands [9], [10]. When structural data for targets is available, molecular docking offers accurate predictions, yet structural data, especially for G-protein coupled receptors (GPCRs), remains limited due to the difficulty in crystallizing such proteins [11], [12], [13]. Recent years have seen the emergence of advanced statistical and machine learning models designed to improve DTI predictions. For example, Yamanishi et al. [14] developed a supervised bipartite graph model that unifies chemical, genomic, and interaction data to predict interactions for four major protein families: enzymes, ion channels, GPCRs, and nuclear receptors. Another model, Predicting Drug Targets with Domains (PDTD) [15], assumes that shared protein domains indicate similar therapeutic potential, emphasizing domain-level interaction prediction. Chen et al. [16] applied a random walk algorithm on a heterogeneous network combining drug similarity, protein similarity, and known interaction data to uncover new DTIs. A hybrid classification approach utilizing Support Vector Machines (SVMs) and Random Forest (RF) also proved effective by incorporating structural and physicochemical properties from protein sequences, successfully identifying novel receptor ligands [17]. Further developments include the use of Restricted Boltzmann Machines (RBMs), as introduced by Zeng and Wang [18], to model a two-layer DTI prediction framework capable of storing diverse interaction types. Wan et al. [19] proposed PDTPS, a method that leverages protein sequence features using Position Specific Scoring Matrices and Bi-gram statistics, applied across four protein families. With the rise of deep learning, these techniques have gained traction for their robustness and superior performance on large-scale biological data. For instance, Deep DTIs, developed by Wen et al. [20], utilize a deep belief network (DBN) to effectively extract features from complex input vectors. Similarly, CGBVS-DNN applies deep neural networks in virtual screening by integrating chemical genomics [21]. Another notable contribution by Wan [22] introduced a deep learning framework capable of learning low-dimensional yet informative features for proteins and compounds using large unlabeled datasets. In this work, we propose a deep learning-based model for DTI prediction that encodes four different protein families and combines descriptors of drugs and protein targets. Drug descriptors are generated using PaDEL, while 115 protein features are obtained via the AAindex1 database [19]. Protein sequences are represented using Moran autocorrelation. After merging the feature vectors, random projection is used to reduce dimensionality to 2916, which is then reshaped into 54×54 matrices.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 gives the concept overview. Then Section 4 presents the proposed model. Section 5 describes the network architecture. Section 6 reports the results, Section 7 present the paper discussion. Section 8 give the methodology used during this paper. and Section 8 concludes the paper.

2. RELATED WORK

Chen et al. [16] utilized an integrated heterogeneous network, including protein-protein similarity network, drug-drug similarity network, and known drug-target interaction networks, by means of the random walk algorithm to predict potential drug-target interactions. A systematic approach based on both Random Forest (RF) and Support Vector Machine (SVM) classifiers took into account the structural and physicochemical properties of proteins derived from primary sequences, which was a robust and efficient tool to distinguish novel scaffold-hopping ligands of the receptors

[17]. Wang and Zeng et al. [18] first designed a two-layer graphical model called a restricted Boltzmann machine (RBM), which made predictions on a multidimensional drug-target network that not only contains binary DTIs but also encodes the types of interactions. A novel method based on protein sequences named PDTPS was proposed, which efficiently applied bi-gram probabilities and Position Specific Scoring Matrix (PSSM) [18]. Further, deep-learning-based methods have gained widespread attention for their capability to address various biological issues. Compared with traditional machine learning methods, deep-learning approaches are more robust, stable, and powerful, especially when dealing with large biological datasets [20]. Wen et al. [20] constructed a predictor named DeepDTIs to infer possible drug-target interactions. This method extracted meaningful representations from massive input vectors through deep-belief networks (DBNs), outperforming other state-of-the-art approaches. Another method, CGBVS-DNN, was proposed as a chemical genomics-based virtual screening technique utilizing deep neural networks to predict compound-protein interactions [21]. Wan [22] proposed a new method to automatically learn low-dimensional but expressive features for compounds and proteins from unlabeled large-scale data, which enhanced the ability to predict DTIs more effectively.

The summary of related work is shown in **Table 1**.

Table 1. The Summary of Related Work

Reference / Work	Method / Approach	Key Features / Details	Remarks
Chen et al. [16]	Heterogeneous network with Random Walk Algorithm	Integrates protein-protein, drug-drug, and known DTI networks	Uses structural and physicochemical properties
Yu et al. [17]	Predicts multiple DTIs from chemical, genomic, and pharmacological data	Systematic, multi-source data integration	Focus on multiple interaction types
Wang and Zeng [18]	Restricted Boltzmann Machine (RBM) for drug-target prediction	Deep graphical model encodes types of interactions	Uses multi-dimensional network data
Meng et al. [19]	Protein sequence-based PDTPS	Uses Bi-gram probabilities and PSSM features	Protein feature engineering
Wen et al. [20]	Deep Belief Network (DeepDTIs)	Extracts representations from massive input vectors	Deep learning approach
Wan [21]	Deep Neural Networks (CGBVS-DNN)	Chemical genomics-based virtual screening	End-to-end deep learning
Wan [22]	Feature Embedding for compounds and proteins	Learns low-dimensional, implicit features	Deep learning with feature embedding
Hinton and Salakhutdinov [23]	Neural Network for Dimensionality Reduction	Reduces data dimensions	Fundamental deep learning technique
He et al. [29]	Predicting networks based on functional groups and biological features	Uses drug functional groups and biological annotations	Network-based prediction Network-based prediction
UniProt Consortium [30]	Protein Knowledgebase	Detailed protein data repository	Data source for protein features
Shaikh et al. [31]	Proteochemometrics + Molecular Docking	Combines chemical and biological data for DTI prediction	Hybrid approach
Yap [32]	Padel Descriptor Software	Calculates molecular descriptors and fingerprints	Tool for feature extraction

Kawashima and Kanehisa [33]	Aaindex: Amino Acid Index Database	Amino acid physicochemical properties	Protein feature database
Li et al. [34]	Profeat Web Server	Computes structural and physico-chemical features of proteins	Protein feature computation tool
Reference / Work	Method / Approach	Key Features / Details	Remarks
Chen et al. [16]	Heterogeneous network with Random Walk Algorithm	Integrates protein-protein, drug-drug, and known DTI networks	Uses structural and physicochemical properties

3. CONCEPT OVERVIEW

The core idea of our proposed system is to utilize deep learning techniques, specifically convolutional neural networks (CNNs), to predict drug-target interactions (DTIs). The approach involves encoding each drug-target pair into a high-dimensional feature vector, which captures relevant chemical and biological information.

The process begins with representing drugs using molecular descriptors calculated via PaDEL-Descriptor software, and proteins are encoded through properties extracted from the AAindex1 database. These properties are further processed using Moran autocorrelation to encode protein sequences effectively.

The feature vectors for drugs and targets are then concatenated to form a combined representation. To manage the high dimensionality and enhance computational efficiency, the combined vectors are projected into a 2916-dimensional subspace using a random projection methodology. These vectors are reshaped into 54×54 matrices, which serve as input to the convolutional neural network.

CNN is designed to learn complex hierarchical features from these matrices. An ensemble of multiple CNN predictors, combined through majority voting, is used to improve prediction robustness and accuracy. The system's architecture and parameters are optimized using validation datasets. This framework allows for the effective capture of nuanced relationships between drugs and targets, leveraging deep learning's capacity for feature extraction from complex biological data. The model is evaluated over various benchmark datasets, demonstrating superior performance compared to traditional machine learning methods

4. PROPOSED MODEL

Random Projection is an effective method that uses a k × d random matrix R to project original d-dimensional data onto a k- dimensional subspace Equation 5. Because of its ease of use and lower rate of inaccurate results as compared to other approaches, it has been used extensively [46]. In this study, a pair's 2939-dimensional vectors were projected onto a 2916-dimensional subspace while roughly maintaining the descriptor distances. The descriptors produced from pharmaceuticals and target proteins (1444 descriptors from drugs and 1495 from targets) do not agree for the concatenated 2939-dimensional descriptors. In other words, every piece of information that is represented for a drug-target pair is not only kept but also standardized to a predetermined scale. Random projections can be used to substitute implicit information, such as protein sequences and the atomic arrangement of chemical compounds. It is actually a data augmentation as well.

Using n distinct random matrices, a deep learning schema was put out to limit the model's capacity for generalization. In order to construct the predictive training model using the CNNs algorithm, each of the n 2916-dimensional drug-target pairs was augmented five times before being reshaped into a 54 × 54 matrix, as illustrated in **Figure 5**. Hidden local information is present in the input vectors even though the CNN's structure is intended to extract local characteristics from images. While the 1D and 2D descriptors of pharmaceuticals capture the interactions among atomic arrangements, the encoding strategy for representing protein sequences takes into account the physicochemical information of nearby amino acids.

All local data was further projected into a low-dimensional subspace using random projection. As a result, as the number of iterations increased, the CNN algorithm progressively recognized deeper and more subtle aspects of the input vectors. Additionally, only the random matrices that produce good results on the validation set and the test set may be used to develop suitable random matrices that could accurately reflect the original feature vector space. The final findings were then generated by the ensemble of prediction scores from the five expanded datasets. If at least

three of the five augmented versions were identified as positive cases, a drug-target pair was projected to interact in this ensemble; if not, the pair was deemed to be non-interacting.

Figure 2 shows how our model pipeline is constructed

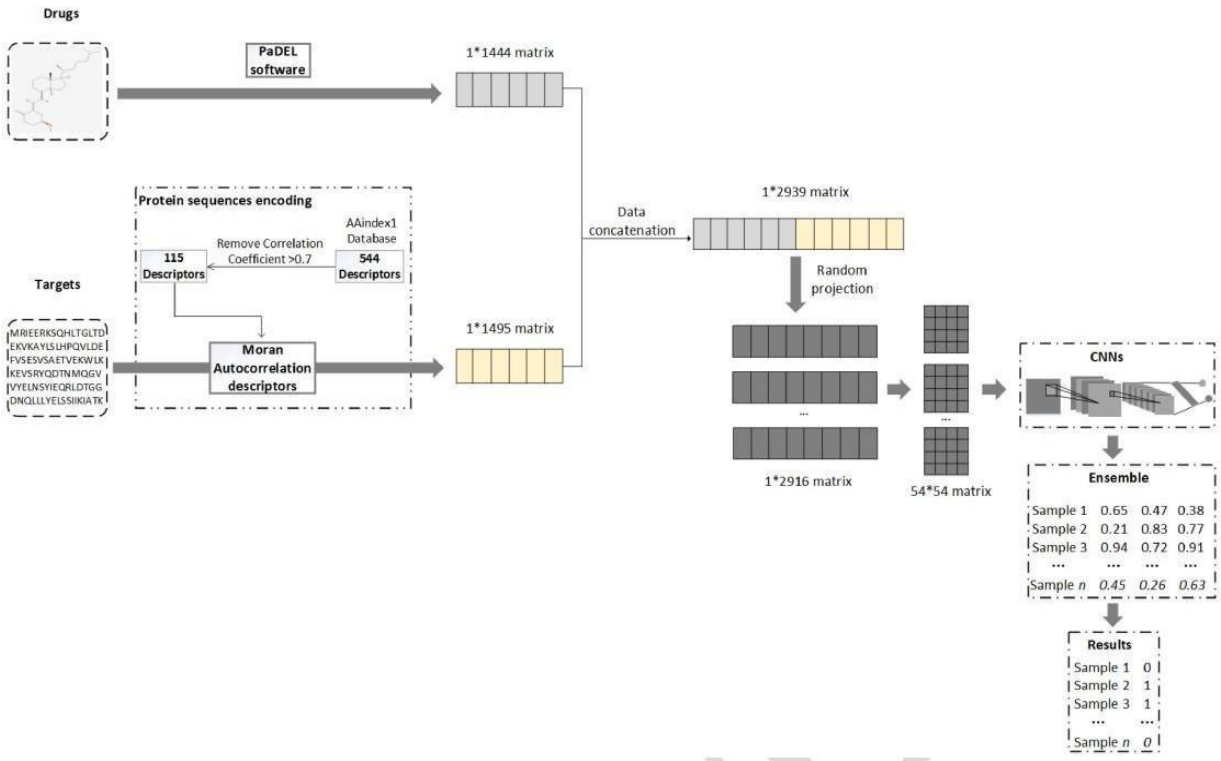

Figure 2. The flowchart of the CNN-based DTI predictions. The final result is represented by 0 or 1, which means non-DTI or DTI.

5. NETWORK ARCHITECTURE

5.1 Benchmark Datasets

Potential interactions between medications and targets were examined using two benchmark datasets, one of which, known as Dataset1, was taken from the KEGG BRITE [27] database;

The other, however, was constructed using the DrugBank [28] database (http://www.drugbank.ca/). While the DrugBank database offers a distinctive bioinformatics and cheminformatics resource with 8261 drug entries with thorough drug and target information, the well-known KEGG BRITE database integrates hierarchical knowledge. Descriptor software was used to construct drug descriptors, and the AAindex1 database was used to extract 115 attributes of each target. Moran autocorrelation was then used to create protein sequences using all of these characteristics. Using random projection methods, the concatenated vectors of drug and target descriptors were projected into a 2916- dimensional subspace. These vectors were then transformed into 54 × 54 matrices, which were used as inputs to train a convolutional neural network (CNN) prediction model. Using a majority vote technique, the ensemble of many predictors on the same drug-target combinations yields the final results.

Different benchmark datasets have been used to evaluate our model. As a result, our methods outperform the state-of-the-art methods on the same benchmark datasets of drugs, genes, and proteins, as well as their interactions and

reactions. Potential interactions between medications and targets were examined using two benchmark datasets, one of which, known as Dataset1, was taken from the KEGG BRITE [27] database. The other, however, was constructed using the DrugBank [28] database (http://www.drugbank.ca/). While the DrugBank database offers a distinctive bioinformatics and cheminformatics resource with 8261 drug entries with thorough drug and target information, the well-known KEGG BRITE database integrates hierarchical knowledge about drugs, genes, and proteins as well as their interactions and reactions.

5.2 Training Dataset

The interaction pairs on Dataset1 retrieved from KEGG BRITE were composed of two parts, one of which was obtained from reference [29]; while the drug-target interactions of the other part were manually collected, where DTIs that overlapped with the first part were omitted. Target proteins were divided into four different categories, including enzymes, ion channels, GPCRs, and nuclear receptors. Protein kinases were integrated into enzymes in the second part. Besides, drugs in the database without structural information and target proteins without primary sequence were removed. In the end, the KEGG BRITE database yielded 5380 known drug-target pairs, made up of 532 targets and 2826 medicines. Supplemental Materials (Supplement_S2), which are accessible online, include a summary of the specific details of the medications, target proteins, and drug-target couples in two sections. Thus, Dataset 1 includes 10177 known target-drug interaction pairs, comprising 1490 targets and 3718 pharmaceuticals, comprising the extensive drug-target interaction entries.

5.3 External Validation Dataset

After removing inorganic or very tiny molecule chemicals from reference [20], an external validation dataset was produced. 1408 approved medications and 1867 non- redundant target sequences associated with these drug entries make up the dataset's 6262 drug target pairs. All of the medications and targets in this dataset differ from those in the training set. The Uniport database (http://www.uniprot.org/), which contains a wealth of protein sequence resources and related comprehensive annotations, contains the target information in this dataset [30].

5.4 Negative Instance Generation

Only experimentally positive DTIs were present in the two databases used in this study. The final performance of DTI prediction models would be significantly impacted by the random selection of unverified non-DTIs because no non-DTI has been empirically validated. Furthermore, there would be significant biases if drug and target properties were combined into a single feature space, particularly for negative cases. A mechanism for extracting reasonably reliable negative examples was established in this work. The following three steps were used to extract negative instances from each of the two datasets: (i) after eliminating the known DTIs, recoupling all medicines and targets in the benchmark dataset into pairs; (ii) ranking all negative occurrences in descending order according to the Euclidean distances between each negative instance and the entire positive set. The negative instance is significantly different from the entire drug-target association space when the Euclidean distance is large. Therefore, even if this hasn't been confirmed in vivo, the drug-target combination is very likely to be non- interaction in theory; (iii) choosing the best ones with as many DTIs as occurrences that are negative. As a result, fewer erroneous non-DTIs were found, and the negative cases were chosen to increase their reliability. DTI positive pairs and non-DTI negative pairs make up the training dataset for our CNN-based approach, as shown in the formula below.

$$V_p = \frac{\frac{1}{L-d}\sum_{i=1}^{L-d}(T_i - \overline{T})(T_{i+d} - \overline{T})}{\frac{1}{L}\sum_{i=1}^{L-d}(T_i - \overline{T})^2}$$

5.5 Drug Descriptors

PaDEL-Descriptor, a free and open-source program that runs on several popular operating systems (Windows, Linux, and MacOS) and can be accessed via a command line interface as well as a graphical user interface, was used to generate the chemical descriptors for chemical compounds [32]. Prior to calculating the descriptors, the molecule's

aromaticity was automatically determined, and salt was eliminated. Since some medications' 3D descriptors are impossible to compute, we only used 1444- dimensional 1D and 2D drug descriptors in this investigation. The formula for a drug candidate is VD = ($VD\ (i)$, i =1, 2, 3, ...,1444).

The following is the method used to encode target proteins. Predictive bias is caused by highly related features among the 544 amino acid physicochemical parameters listed in the AAindex1 database [33]. This was accomplished by ranking the number of pertinent properties in descending order and calculating the correlation coefficient of each property with the others. As a result, every associated property from the top one was gradually removed. Until there were no more pairs of attributes with a correlation coefficient greater than 0.7, the procedure was repeated. Lastly, 115. After obtaining attributes, autocorrelation descriptors were utilized to encode protein sequences.

The distribution of amino acid characteristics serves as the basis for Moran autocorrelation descriptors [34], [35], which consider neighbor information about amino acids and are defined as follows:

where Ti and Tiþd are the property values for residues I and Iþ d, respectively; d is the distance between the ith residue and the nearby residue i þ d; Based on the training results of our model in this study, d is set to 13; T is the average value of Ti T=(), i.e., T = ($\sum_{i=1}^{L} T_i$) / L; and L is the length of the protein sequence.

5.6 protein representation

In order to encode a single target protein, 13-dimensional vectors are created for each attribute. One target protein is characterized by concatenating all of the resultant vectors after the calculation is applied to each of the 115 attributes. In other words, each target protein in our experimental work is represented by a 1495-dimensional vector, which may be created by VP = (VP(j); j = 1, 2, ..., 1495). In this manner, the method for encoding target proteins preserves high protein sequence information in addition to adequate physicochemical information about amino acids, as shown in **Figure 3**.

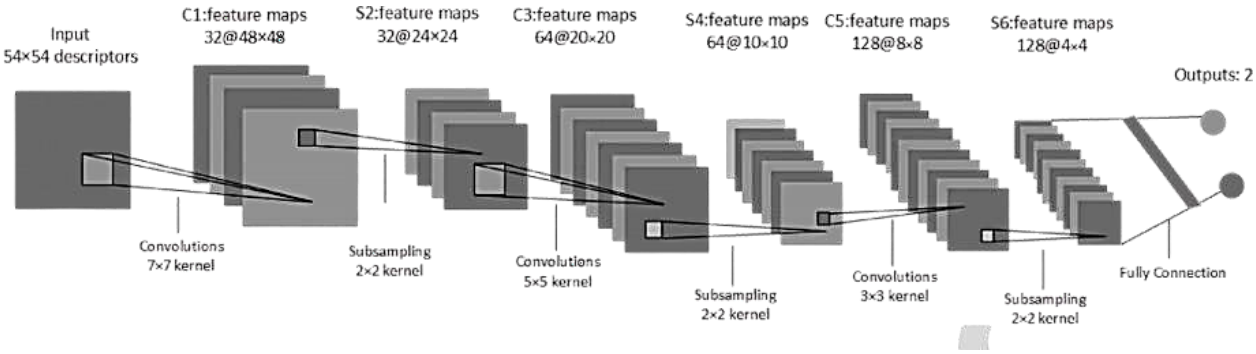

Figure 3. Shows the Model Representation

5.7 Construction of Datasets

For each drug-target pair, let V = (VD(i), VP(j)), where i = 1, 2, ..., 1444 and j = 1, 2,..., 1495. The variable T = [T_1, T_2, T_3, ..., T_{1495}] represents the protein descriptors. Each pair consists of a 1444-dimensional vector of drug descriptors (VD) and a 1495- dimensional vector of protein descriptors (VP). By concatenating these two vectors, a final 2939-dimensional vector V = (VD(i), VP(j)) ∈ $\mathbb{R}^{1x2939}$ is obtained, which characterizes each drug-target pair.

5.8 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a subset of deep learning architectures that require far fewer parameters for training than conventional artificial neural networks that are fully connected [36]. As a result, CNNs are widely used in fields like natural language processing, recommender systems, and image and video recognition. Convolutional layers, pooling layers, and fully connected layers are the three main components of a CNN's topology [37].

The fundamental components of the entire system are convolutional layers. By sliding convolution kernels on top of earlier layers, deeper convolutional layers typically pick up more expressive features. The size of layer outputs is controlled by several hyperparameters, including the depth and size of the filters, the distance between filters, and the number of zero-paddings surrounding the input feature map [38], [39]. By using local non-linear processes, pooling layers can guarantee translation invariance and minimize the amount of input features [40]. The final few layers of CNNs usually have fully connected layers, which have an equal number of out-Consider neurons to be conventional neural networks. In particular, CNNs and LeNet-5 [41], a type of CNN intended for handwritten and machine-printed character recognition, share a similar structure. **Figure 3** displays the three convolutional layers, three max-pooling layers, and one fully connected layer that make up our model's architecture.

Even with the addition of an additional convolutional and max-pooling layer in comparison to LeNet-5, the model saw a significant improvement, albeit at the cost of a slightly longer program runtime and increased computational complexity.

The discussion section contains the results.

For each convolutional network layer, the outputs of the layer are calculated by the following formula:

$$y_k^l = f\left(\sum_m W_{m,k}^l y_m^{l-1} + b_k^l\right),$$

Where l is the layer index, m is the index of input feature maps, and k is the index of output feature maps. The input y_k^l refers to the k-th feature map in layer l, while the output y_m^{l-1} refers to the m-th feature map from layer l - 1. Here, W represents the convolutional weight tensor, and b is the bias term. The activation function f(.) is a nonlinear element-wise function applied to each feature value. In our model, we use the Rectified Linear Unit (ReLU) activation function, which helps reduce computational cost and mitigate the vanishing gradient problem during backpropagation training [42].

The output layer of our model is a logistic regression classifier. It takes y_k^l as input and computes the final prediction score using the following expression:

$\hat{y} = f(W^l y^l + b^l)$

In this equation, $\hat{y}$ is the final predicted score, W is the weight matrix, and b is the bias vector. The output is a 2-dimensional vector representing positive and negative classes for the binary classification task of drug–target interaction (DTI) prediction.Training and Optimization:To train the model and find optimal parameters, we minimize the cross-entropy loss using the Adaptive Moment Estimation (Adam) optimization algorithm [43], along with backpropagation. The loss function is given by:

$Loss = -(1/N) \Sigma [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)],$

Where,

N is the number of training samples, y_i is the true label, and $\hat{y}_i$ is the predicted probability for the i-th sample.

Regularization Techniques: To further improve the model's performance, we apply two techniques: Dropout [44]. This technique randomly drops some units in the fully connected layers during training to prevent overfitting by reducing co- adaptation of neurons. Batch Normalization [45]: This technique standardizes the inputs of each layer to have zero mean and unit variance, which improves training stability and allows the use of higher learning rates.

6. EXPERIMENTAL RESULTS

Assessment of DTI Performance. The entire dataset was divided into three sections—the training set, validation set, and test set, with respective ratios of 0.7, 0.1, and 0.2 in order to estimate the performance of our model. The model was pre-trained and fine-tuned using the training set; its parameters were optimized using the validation set; and the model was evaluated using the test set. The model was run ten times to produce consistent and trustworthy results, and Table 4 shows the average prediction performance. The accuracy, sensitivity, precision, and F1 score of the model on the validation set are 0.9818, 0.97701, 0.9948, and 0.9823, respectively; on the test set, the results are 0.9800, 0.9713, 0.9887, and 0.9798. In the dataset, it is seen that our model produces greater precision than sensitivity,

indicating that it is very capable of detecting false positives. It is generally known that high false positive instances have an impact on the identification performance of DTIs using other computational methods. In the DTI's prediction, it would be suggested that a strong CNN-based predictive model, along with an efficient encoding approach for targets and 2D descriptors for pharmaceuticals, is reliable, accurate, and efficient. Our model achieved impressive AUC values of 0.9966 and 0.9965 on the validation set and test set, respectively, as illustrated in **Table 2**. The ROC curve indicates that the model is performing well. It makes it abundantly evident that our suggested model, which has a high true-positive rate compared to a low false-positive rate, can detect true DTIs by capturing adequate and effective features using a deep learning approach.

Table 2. The Detailed Prediction Performance on the Validation Set and Test Set of Dataset1, Respectively

Methods	Acc	Sen	Pre	F1	AUC
Validation set	0.9818	0.9701	0.9948	0.9823	0.9934
Test set	0.9800	0.9713	0.9887	0.9798	0.9965

Comparing Different Machine Learning Techniques. We compared the suggested method with the most advanced machine learning techniques based on Dataset1 to further demonstrate its effectiveness and robustness. The performance comparison of Gaussian Naive Bayesian (GBN), k-Nearest Neighbor (KNN), random forests, and our suggested approach is shown in **Table 3**.

Every classifier produces the best results in **Table 3**.

Table 3. Performance Comparison with Other Machine Learning Methods on Dataset1

Method	Acc ±SD	Sen ±SD	Pre ±SD	F1 ±SD	AUC ±SD
Our model	0.9800±0.0066	0.9713±0.0172	0.9887±0.0238	0.9798±0.0073	0.9965±0.0019
KNN	0.8302±0.0127	0.8517±0.0324	0.7028±0.0385	0.7704±0.0142	0.8966±0.0087
Random Forests	0.9117±0.0105	0.8517±0.0324	0.8778±0.0317	0.8673±0.0103	0.9579±0.0059
GBN	0.8002±0.0129	0.8947±0.0203	0.7520±0.0362	0.8172±0.0124	0.8742±0.0093

To evaluate generalization, the model was compared with a DBN-based approach using two datasets: Dataset 2, randomly selected negative samples, the same as those used in the literature. Dataset 3: an equal number of positive and negative samples, as shown in **Table 4**.

Table 4: Overall Performance Comparison of DBN and Our Proposed Method on Dataset2 and Dataset3

Methods	TPR ±SD	TNR ±SD	Acc ±SD	AUC ±SD
DBN	0.8227±0.0065	0.8953±0.0130	0.8588±0.0049	0.9158±0.0059
Our model (Dataset2)	0.9194±0.0091	0.9114±0.0196	0.8814±0.0075	0.9527±0.0043
Our model (Dataset3)	0.9709±0.0067	0.9709±0.0067	0.9604±0.0032	0.9947±0.0021

By adjusting the parameters. It has been noted that every classifier produced accurate results. Out of the four classifiers, our model performs the best; in particular, it outperforms RF and KNN classifiers in terms of accuracy (6.8% higher than RF and 15% higher than KNN). Furthermore, a more significant evaluation metric that gauges the harmony between precision and sensitivity rates is the F1 score. Despite the small size of the DTIs dataset, our method has the highest F1 score (0.9798) out of the four machine learning techniques, demonstrating that CNN's algorithm outperforms conventional machine learning techniques in inferring drug-target associations. In order to supplement the DTI's classification, the deep learning technique is thought to be an efficient means of extracting informative features. However, this methodology offers a means of identifying the relationship between DTIs and is anticipated to enhance the prediction performance of DTIs. The satisfactory performance may be due to the efficient and trustworthy selection of negative instances. The matrices serve as inputs to a convolutional neural network (CNN) architecture. A majority voting ensemble of multiple predictors yields the final interaction predictions as shown in **Figure 1**.

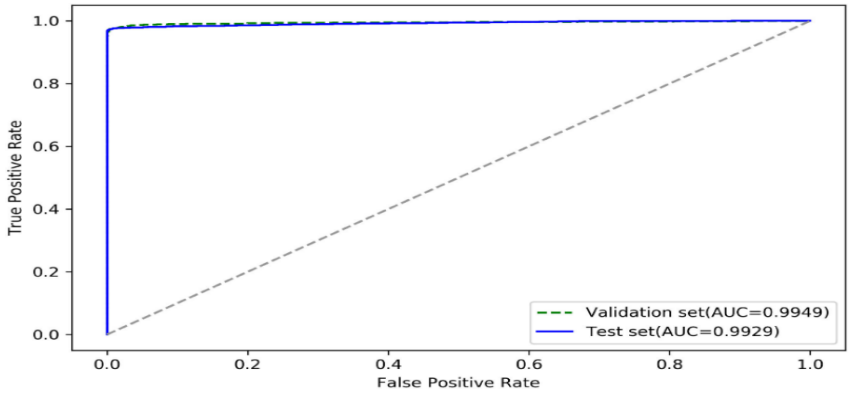

Figure 1. ROC curves of our method on both the validation set and test set.

We evaluated our model on several benchmark datasets, where it consistently outperformed current state-of-the-art methods.

Deep-Belief Network Comparison

To further evaluate our model's capacity for generalization, an external validation dataset that is non-redundant with the training dataset and obtained from another source is then tested. Deep learning-based methods are currently being used extensively in various biological domains to identify drug-target relationships. Convolutional neural networks were less frequently utilized in the categorization of biological data, but the majority of studies concentrated on deep-belief networks, which were proposed by Hinton et al. [23], [24]. Another type of deep learning technique is DBN, which is greedily trained and consists of many stacked RBMs. A greedy layer-wise unsupervised training process and a supervised fine-tuning process are the two successive processes that make up this model. To infer drug-target relationships in literature, a DBN-based algorithm framework called Deep DTIs was created [20]. However, random selection was utilized to obtain negative instances of the dataset required to train Deep DTI's model, which other efforts would not be able to repeat. The experiments of our work randomly picked the same number of negative cases as the literature [20], known as Dataset2, in order to ensure a fair comparison. In order to create Dataset 3, we also chose negative examples in equal numbers to the positive examples. Our model's performance on Dataset 2 yielded accuracy and AUC of 0.8814 and 0.9527, as indicated in **Table 5**. which, respectively, are 2.26 and 3.69 percent greater than the baseline approach. In the meantime, selecting negative examples from Dataset 3 instead of selecting them at random significantly improves prediction performance in DTIs. The enhancement of our model's performance further shows that it could work with various DTI dataset resources.

7. DISCUSSION

7.1 Hyper-Parameter Adjustment

The three hyperparameters of the suggested CNN model, the batch normalization layer, the learning rate, and the neural network topology were the primary emphasis of this section. To investigate the ideal parameters, we only changed one hyperparameter while fixing the others.

7.2 Neural Networks Topology

Deeper CNNs perform better than lower CNNs because deeper neural networks are able to extract more valuable information. But in order to train, deep CNNs need a lot of parameters, which takes more time and equipment. Deeper neural networks improve accuracy even more than the baseline does. On the training dataset, the topology of the CNNs was contrasted with that of LeNet 5. LeNet-5 did, in fact, perform marginally worse than our model, obtaining an Acc of 0.9654, a Sen of 0.9536, a Pre of 0.9766, an F1 of 0.9650, and an AUC of 0.9643. With one additional convolutional layer and one additional pooling layer, our model's accuracy is 0.980, demonstrating that a model with a high number of convolutional layers may extract more valuable information, making it more resilient and effective.
Accuracy and convergence are strongly impacted by the learning rate:

- Informative features are not learned by a high learning rate (e.g., 0.01).

- The optimal performance is achieved at a low learning rate (1e-4).
- The performance of various learning rates is shown in **Table 5**.

Tab. 5. Performance of Our Model with Different Learning Rates

parameters	Acc	Sen	Pre	F1	AUC
1e-2	0.500	0.853	0.806	0.829	0.942
1e-3	0.9776	0.9750	0.980	0.9775	0.9960
1e-4	0.9800	0.9713	0.9887	0.9798	0.9965
1e-5	0.9758	0.9641	0.9872	0.9755	0.9946
1e-6	0.923	0.9157	0.9293	0.9221	0.9234

7.3 Learning Rate

The CNN's topology is significantly influenced by the learning rate. A higher learning rate causes CNNs' gradients to drop more quickly, which prevents the extraction of significant and instructive features as the number of iterations increases. Small ones, on the other hand, cause CNNs to scarcely converge and require more time to train the entire prediction model. As a result, five distinct learning rate scales between 1e-2 and 1e-6 were examined for the suggested model. With an accuracy of 0.5, this model showed that relevant features cannot be learned to identify DTIs when the learning rate is set to 0.01. However, the model performs powerfully with an accuracy of more than 0.97 when the learning rate is between 1e-3 and 1e-6 **Figure 4**; performance is marginally better when the learning rate is between 1e-4.

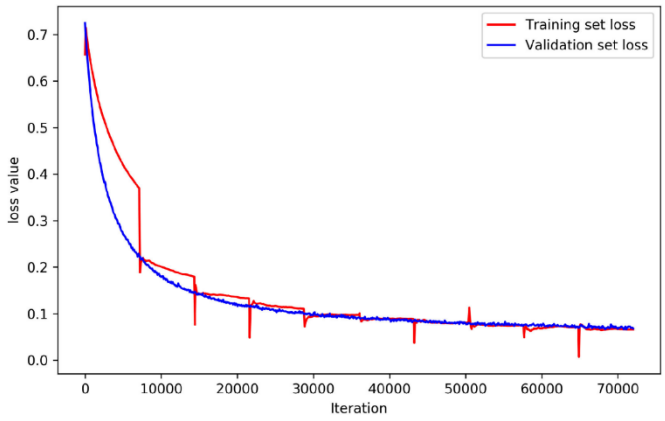

Figure 4: The variation tendency of loss function curves on the training set and validation set, respectively

7.4 Layer of Batch Normalization

Batch normalization (BN) is a new and useful technique to boost classifier performance. This study has demonstrated unequivocally how training is facilitated by a batch normalization layer that normalizes the input data. The accuracy of the model with the BN layer is about 4% greater than that of the model without BN, even when both learning rates are kept constant. In order for our model to predict input instances as either interaction or non-interaction, it demonstrates that BN can successfully normalize data into the opposite range.

7.5 Loss Value

One crucial statistic for assessing neural network convergence is loss value. The training and validation sets' respective loss variation 10 dency is displayed in Fig. 4. During the first 15,000 iterations, losses fluctuated quickly, resulting in a precipitous decline in losses. But as the number of repetitions increases, it gradually declines. After 40000 iterations, the loss value finally varies within a narrow range of 0.06. It should be noted that at the conclusion of iterations, the loss value on the validation set is nearly equal to that on the training set, indicating that our model is robust to detecting DTIs between medicines and target proteins and has a fast convergence speed.

7.6 Other

Two aspects of our experiments, data augmentation and different matrix scales, are covered in this section. Our tests have shown that data augmentation can produce reliable and consistent outcomes. A competent deep learning-based

model can really learn more valid and appealing features to attain adequate and believable performance with the use of vast amounts of data. Regarding different matrix scales, slightly large-scale dimensions of data make the model more likely to achieve better results at the cost of increasing computation complexity. So the dimension of data was reshaped to 54 x 54 in this work, which not only has slightly influence on final predictions but also costs less calculation resources.

8. METHODOLOGY

8.1 Data Collection and Benchmark Datasets

This study follows a comprehensive and structured methodology designed to predict drug-target interactions (DTIs). Two primary benchmark datasets were utilized for this research. The first dataset was derived from KEGG BRITE, consisting of validated DTIs curated through a combination of published literature and manual annotation. The second dataset was obtained from DrugBank, providing a broader and more diverse collection of drug-target pairs. Both datasets included positive instances (confirmed DTIs) and incorporated a careful strategy for generating reliable negative samples to support robust model training.

8.2 Negative Instance Generation

To improve the credibility of negative samples and avoid the risk of including false negatives a common issue in random negative sampling, a systematic technique was applied. All possible drug-target combinations, excluding known DTIs, were generated. These candidate negative pairs were then ranked based on their Euclidean distances from positive pairs, calculated using feature descriptors. Pairs with the largest distances, indicating a lower likelihood of representing true interactions, were selected. The number of selected negative instances was balanced to match the number of positive DTIs, ensuring a clean and reliable training dataset that minimized noise and improved learning quality.

8.3 Feature Extraction

Feature descriptors were independently extracted for both drugs and target proteins. For drugs, PaDEL-Descriptor software was employed following preprocessing operations, such as salt removal and aromaticity identification. This produced a 1444-dimensional feature vector composed of 1D and 2D molecular descriptors for each drug.

For target proteins, physicochemical and biochemical property values were retrieved from the AAindex1 database. Each protein sequence was encoded into 115 properties, and the sequences were processed using Moran autocorrelation to capture correlations of amino acid properties along the sequence, enhancing the representation of the protein's structural and functional characteristics.

8.4 Feature Concatenation, Projection, and Reshaping

The extracted drug and protein feature vectors were concatenated to form a unified feature vector for each drug-target pair. To manage high dimensionality and maintain computational efficiency, random projection was applied, reducing the original ~2939-dimensional feature vectors to a 2916-dimensional subspace while preserving their relative pairwise distances. This transformation also introduced implicit data augmentation by encoding additional structural variance through random mapping. The projected feature vectors were then reshaped into 54×54 matrices, formatted to serve as suitable input for a Convolutional Neural Network (CNN).

8.5 Deep Learning Model Construction

The predictive model employed in this study was a Convolutional Neural Network (CNN) inspired by the classic LeNet-5 architecture. The network consisted of three convolutional layers, each followed by pooling layers, designed to progressively capture hierarchical feature representations. A fully connected layer at the end of the network aggregated these features for the final prediction. This architecture allowed the model to automatically learn discriminative and abstract features from the structured input matrices, eliminating the need for manual feature engineering.

8.6 Training Procedure

Model training was guided by a cross- entropy loss function, which quantified discrepancies between predicted and actual class labels. The Adam optimizer was used for updating network weights via backpropagation. To enhance model stability and prevent overfitting, dropout regularization was applied to the fully connected layers by randomly deactivating units during training. Additionally, batch normalization was incorporated to standardize the inputs of each

layer, facilitating faster and more stable convergence. Key hyperparameters, including learning rate, dropout rate, and CNN architecture configurations, were experimentally tuned to maximize the model's performance.

8.7 Model Evaluation

The dataset was divided into training, validation, and testing subsets in a 70%, 10%, and 20% split, respectively. The model underwent multiple training iterations to assess stability and consistency, with average values for performance metrics such as accuracy, sensitivity, precision, F1- score, and area under the ROC curve (AUC) reported across runs. To evaluate the generalizability of the trained model, external validation was conducted using an independent dataset derived from DrugBank, ensuring reliable performance assessment beyond the original training data.

8.8 Ensemble Learning Strategy

To further enhance classification robustness and reduce model variance, an ensemble learning strategy was implemented. Multiple independently trained CNN models were aggregated through a majority voting mechanism, where the final classification for each drug-target pair was determined based on the most frequently predicted label from the ensemble members. This ensemble approach improved predictive accuracy by leveraging the complementary strengths of different model instances.

9. Conclusions

In this study, a convolutional neural network-based method was developed to predict drug-target interactions (DTIs). The model effectively learned deep features from encoded drug and target data, demonstrating high accuracy, sensitivity, precision, and AUC on benchmark datasets [20], [23], [43]. The results indicated that the CNN approach outperformed traditional machine learning algorithms like KNN, Random Forest, and GBM, especially in accuracy and F1 score, highlighting its robustness and capability in DTI prediction. Moreover, the model showed strong generalization ability on external datasets and benefited significantly from careful negative sample selection. Overall, the proposed deep learning framework offers a promising tool for accelerating drug discovery and understanding drug mechanisms, with potential for further improvements through integrating more biological information and optimizing hyperparameters [44], [45].

Acknowledgments

The authors sincerely thank the referees, Associate Editor, and Editor-in-Chief for their valuable comments and suggestions, which have greatly improved this paper. The authors also acknowledge the use of DeepSeek for assistance in improving the English grammar and language clarity.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Reference

[1] L. Wang, Z. H. You, X. Chen, X. Yan, G. Liu, and W. Zhang, "Rfdt: A rotation forest-based predictor for predicting drug-target interactions using drug structure and protein sequence information," Current Protein Peptide Science, vol. 19, pp. 445–454, 2016.
[2] B. Booth and R. Zemmel, "Prospects for productivity," Nature Reviews Drug Discovery, vol. 3, no. 5, pp. 451–456, 2004.
[3] Y.-A. Huang, Z.-H. You, and X. Chen, "A systematic prediction of drug-target interactions using molecular fingerprints and protein sequences," Current Protein & Peptide Science, vol. 19, pp. 468–478, 2018.
[4] A. L. Hopkins, "Network pharmacology: The next paradigm in drug discovery," Nature Chemical Biology, vol. 4, no. 11, pp. 682–690, 2008.
[5] M. J. Keiser, B. L. Roth, B. N. Armbruster, P. Ernsberger, J. J. Irwin, and B. K. Shoichet, "Relating protein pharmacology by ligand chemistry," Nature Biotechnology, vol. 25, no. 2, pp. 197–206, 2007.
[6] R. Iorio, R. Shrestha, M. Berube, and A. R. Licinio, "Pathway analysis of polypharmacology," Briefings in Bioinformatics, vol. 15, no. 2, pp. 278–289, 2014.

[7] D. Lounkine, M. J. Keiser, S. Whitebread, D. Mikhailov, J. Hamon, J. L. Jenkins, P. Lavan, E. Weber, A. K. Doak, S. Côté, B. K. Shoichet, and L. Urban, "Large-scale prediction and testing of drug activity on side-effect targets," Nature, vol. 486, pp. 361–367, 2012.
[8] M. Campillos, M. Kuhn, A.-C. Gavin, L. J. Jensen, and P. Bork, "Drug target identification using side-effect similarity," Science, vol. 321, no. 5886, pp. 263–266, 2008.
[9] X. Chen, H. Y. Ji, G. Y. Yan, and L. Y. Han, "Drug–target interaction prediction: databases, web servers and computational models," Briefings in Bioinformatics, vol. 17, no. 4, pp. 696–712, 2016.
[10] Z. Li, P. Han, and J. Lin, "A machine learning based method for predicting drug–target interactions using drug fingerprints and protein sequence descriptors," Molecular BioSystems, vol. 12, no. 7, pp. 2431–2439, 2016.
[11] B. R. Donald, Algorithms in Structural Molecular Biology, Cambridge, MA, USA: MIT Press, 2011.
[12] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson, "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility," Journal of Computational Chemistry, vol. 30, no. 16, pp. 2785–1795, 2009.
[13] A. C. Cheng, R. G. Coleman, K. T. Smyth, Q. Cao, P. Soulard, D. R. Caffrey, A. C. Salzberg, and E. S. Huang, "Structure-based maximal affinity model predicts small-molecule druggability," Nature Biotechnology, vol. 25, no. 1, pp. 71–75, 2007.
[14] Y. Yamanishi, M. A. Araki, W. Honda, and M. Kanehisa, "Prediction of drug-target interaction networks from the integration of chemical and genomic spaces," Bioinformatics, vol. 24, no. 13, pp. i232–i240, 2008.
[15] Y.-Y. Wang, J. C. Nacher, and X.-M. Zhao, "Predicting drug targets based on protein domains," Molecular Biosystems, vol. 8, pp. 1528–1534, Apr. 2012.
[16] X. Chen, M. X. Liu, and G. Y. Yan, "Drug-target interaction prediction by random walk on the heterogeneous network," Molecular BioSystems, vol. 8, no. 7, pp. 1970–1978, 2012.
[17] H. Yu, J. Chen, X. Xu, Y. Li, H. Zhao, Y. Fang, X. Li, W. Zhou, W. Wang, and Y. Wang, "A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data," PLoS ONE, vol. 7, no. 5, 2012.
[18] Y. Wang and J. Zeng, "Predicting drug-target interactions using restricted Boltzmann machines," Bioinformatics, vol. 29, no. 13, pp. 126–134, 2013.
[19] F.-R. Meng, Z.-H. You, X. Chen, Y. Zhou, and J.-Y. An, "Prediction of drug-target interaction networks from the integration of protein sequences and drug chemical structures," Molecules, vol. 22, no. pii: E1119, Jul. 2017.
[20] W. Ming, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, and H. Lu, "Deep-learning-based drug-target interaction prediction," Journal of Proteome Research, vol. 16, no. 4, 2017.
[21] M. Hamanaka, K. Taneishi, H. Iwata, J. Ye, J. Pei, J. Hou, and Y. Okuno, "Cgbvsdnn: Prediction of compound-protein interactions based on deep learning," Molecular Informatics, vol. 36, no. 1/2, 2017.
[22] F. Wan and J. Zeng, "Deep learning with feature embedding for compound-protein interaction prediction," 2016.
[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, pp. 504–507, Jul. 2006.
[24] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.
[27] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "KEGG as a reference resource for gene and protein annotation," Nucleic Acids Research, vol. 44, no. D1, pp. D457–D462, 2016.
[28] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, and A. Assempour, "DrugBank 5.0: a major update to the DrugBank database for 2018," Nucleic Acids Research, vol. 46, no. D1, pp. D1074–D1082, 2018.
[29] Z. He, J. Zhang, X. H. Shi, L. L. Hu, X. Kong, Y. D. Cai, and K. C. Chou, "Predicting drug-target interaction networks based on functional groups and biological features," PLoS ONE, vol. 5, no. 3, 2010.
[30] T. U. Consortium, "UniProt: The universal protein knowledgebase," Nucleic Acids Research, vol. 45, no. D1, pp. D158–D169, 2017.
[31] N. Shaikh, M. Sharma, and P. Garg, "An improved approach for predicting drug-target interaction: Proteochemometrics to molecular docking," Molecular BioSystems, vol. 12, no. 3, 2016.
[32] C. W. Yap, "Padel-descriptor: An open-source software to calculate molecular descriptors and fingerprints," Journal of Computational Chemistry, vol. 32, no. 7, pp. 1466–1474, 2011.
[33] S. Kawashima and M. Kanehisa, "Aaindex: Amino acid index database," Nucleic Acids Research, vol. 27, no. 1, pp. 368–369, 1999.
[34] Z. R. Li, H. H. Lin, L. Y. Han, L. Jiang, X. Chen, and Y. Z. Chen, "Profeat: A web server for computing structural and physico-chemical features of proteins and peptides from amino acid sequence," Nucleic Acids Research, vol. 39, no. Web Server issue, 2011.

[35] Z. R. Li, L. Y. Han, L. Jiang, X. Chen, and Y. Z. Chen, "Prediction of subcellular location of mycobacterial proteins using feature fusion and support vector machine," Journal of Proteome Research, vol. 5, no. 11, pp. 2780–2788, 2006.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.
[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations (ICLR), 2015.
[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034, 2015.
[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," European Conference on Computer Vision (ECCV), Springer, pp. 818–833, 2014.
[41] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," Advances in Neural Information Processing Systems (NeurIPS), vol. 2, pp. 396–404, 1990.
[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 807–814, 2010.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," International Conference on Learning Representations (ICLR), 2015.
[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 448–456, 2015.
[46] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson–Lindenstrauss lemma," International Computer Science Institute Technical Report, 1999