Deep learning in target prediction and drug repositioning: Recent advances and challenges

Jun-Lin Yu[1], Qing-Qing Dai[1], Guo-Bo Li*

Key Laboratory of Drug-Targeting and Drug Delivery System of the Education Ministry and Sichuan Province, Sichuan Engineering Laboratory for Plant-Sourced Drug and Sichuan Research Center for Drug Precision Industrial Technology, West China School of Pharmacy, Sichuan University, Chengdu 610041, China

Drug repositioning is an attractive strategy for discovering new therapeutic uses for approved or investigational drugs, with potentially shorter development timelines and lower development costs. Various computational methods have been used in drug repositioning, promoting the efficiency and success rates of this approach. Recently, deep learning (DL) has attracted wide attention for its potential in target prediction and drug repositioning. Here, we provide an overview of the basic principles of commonly used DL architectures and their applications in target prediction and drug repositioning, and discuss possible ways of dealing with current challenges to help achieve its expected potential for drug repositioning.

Keywords: Deep learning; Drug repositioning; Target prediction; Drug–target interaction; Heterogeneous network; Drug discovery

Introduction

Drug repositioning is becoming an increasingly attractive direction in drug discovery and development because it involves potentially shorter development timelines and lower development costs compared with target-centric *de novo* drug discovery and development.[1,2] Success stories of drug repositioning are increasing in number, which historically has been largely opportunistic and serendipitous.[1,2] For example, sildenafil, originally developed as an antihypertensive drug, was repurposed for erectile dysfunction mainly based on retrospective clinical experience; thalidomide was repurposed by serendipity for multiple myeloma erythema and erythema nodosum leprosum.[2] In recent decades, the development of relatively systematic approaches has facilitated the identification of the right candidate for a given indication. Notably, during the coronavirus disease 2019 (COVID-19) pandemic, several non-antiviral drugs, such as leflunomide[3,4] and baricitinib,[5] have been repurposed by various approaches as new treatments to participate in the battle against COVID-19.[6,7] Systematic drug repositioning approaches can be categorized into computational approaches and experimental approaches, both of which are frequently being used together. Computational approaches offer a relatively quick and resource-saving way to identify testable hypotheses that promote drug repositioning.[8,9]

Various computational approaches have been established for drug repositioning, including structure-based, ligand-based, and data-driven approaches. Chen et al. pioneered the use of inverse molecular docking, a structure-based approach, named INVDOCK,[10] for target prediction, leading to the development of several upgraded structure-based methods and web servers, such as TarFisDock,[11] VinaMPI,[12] idTarget,[13] and iRAISE.[14] These methods have been successfully used for target prediction and drug repositioning; for example, Dakshanamurthy et al. discovered the antiparasitic drug mebendazole as a new inhibitor for vascular endothelial growth factor receptor 2, an anticancer drug target.[15] Ligand chemical similarity has also been frequently used

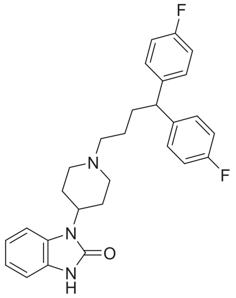
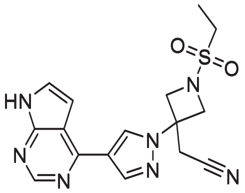
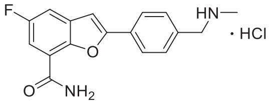
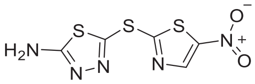
* *Corresponding author.Li, G.-B. (liguobo@scu.edu.cn)*

[1] Co-first authors.

1359-6446/© 2021 Elsevier Ltd. All rights reserved.
https://doi.org/10.1016/j.drudis.2021.10.010

TABLE 1

Examples of successful drug repositioning using DL approaches.

Approved or investigational drug	Original indication (target)	New indication (target)	Method	Comment	Refs
 Pimozide	Tourette syndrome (dopamine D2 and D3 receptor antagonist)	NSCLC	DNN	Li et al. established a drug-repurposing approach based on transcriptomic data and chemical structures using DNN and identified Pimozide as a possible candidate for treating NSCLC	29
 Baricitinib	Rheumatoid arthritis (Janus kinase inhibitor)	COVID-19 (AP2-associated protein kinase 1 and cyclin G-associated kinase)	DNN	Stebbing et al. reported baricitinib as potential treatment for 2019-nCoV acute respiratory disease	30
 Mefuparib (CVL218)	Anticancer (poly-ADP-ribose polymerase 1 inhibitor, PARP1)	COVID-19 (possibly binds to N-terminal domain of nucleocapsid protein)	GCN	Ge et al. indicated that mefuparib exhibited effective inhibitory activity against SARS-CoV-2 replication and had anti-inflammatory effects	31
 Halicin (SU3327)	Diabetes (c-Jun N-terminal kinase inhibitor)	Broad-spectrum bactericidal antibiotic (dissipating proton motive force)	D-MPNN	Stokes et al. discovered halicin as a new bactericidal antibiotic against a phylogenetic spectrum of pathogens by using trained D-MPNN network	28

for target prediction. Keiser et al. proposed a similarity ensemble approach to quantitatively group related protein targets based on their ligand similarity encoded by Extended Connectivity Fingerprint (ECFP).[16,17] This approach has resulted in several new drug-target associations, demonstrating its effectiveness.[17] Several 3D similarity methods, such as ChemMapper[18] and Gaussian Ensemble Screening,[19] have been established for target prediction and drug repositioning. Pharmacophore, as an alternative to the representation of essential features for a ligand binding with its specific target receptor, has also been widely used for target prediction and drug repositioning.[20] PharmMapper is the most representative approach, which was developed as a user-friendly, freely accessed web-server.[21,22] By considering target-specific features, a protein–ligand interaction fingerprinting-based method, named IFPTarget,[23] was developed that showed improved prediction ability to retrieve known targets, and has been successfully used to repurpose colchicine for targeting glycine receptor alpha 3.[23,24] Data-driven approaches have been increasingly established for drug repositioning, for example, based on clinical phenotypes, adverse event profiles, cheminformatics, transcriptomic, proteomics or metabolomics data.[25,26] For example, Campillos et al. proposed a phenotypic side-effect similarity approach for drug repositioning, and experimentally validated its feasibility to infer new molecular interactions and exploit new uses of marketed drugs.[25]

DL is a fast-growing subfield of artificial intelligence (AI), which uses various artificial neural network (NN) algorithms that mimic human brain neurons to learn high-level abstractions contained in data. Recent advances in DL methods have boosted the evolution of drug discovery.[27] Given their powerful learning ability, they can effectively capture high-level hidden representations from various raw data, including heterogeneous data, and show a state-of-the-art performance. Compared with traditional machine-learning (ML) approaches, such as support vector machine (SVM) and random forest (RF), DL approaches are capable of learning more abstract information by building deep architectures without manually selecting and tuning features. Several successful drug repositioning cases were reported recently (Table 1).[28–31] For example, Li et al. presented a new deep NN (DNN) with transcriptomic data and chemical structural information, and identified pimozide, an antidyskinesia agent, as a potential candidate for nonsmall cell lung cancer (NSCLC). Stokes et al. adopted a directed message-passing NN (D-MPNN) architecture capable of predicting compounds with antibacterial activities, and successfully repurposed halicin (SU3327),[28] a c-Jun N-terminal kinase inhibitor, as a new bactericidal antibiotic effectively against various Gram-negative pathogens, including *Acinetobacter baumannii*, highlighting the utility of DL approaches to exploit new space for drug repositioning.

In this review, we provide an overview of the basic understanding of commonly used DL architectures. Subsequently, we summarize drug–target interaction (DTI)- and heterogeneous network-based DL approaches for target prediction and drug repositioning, and highlight their specific features, strengths

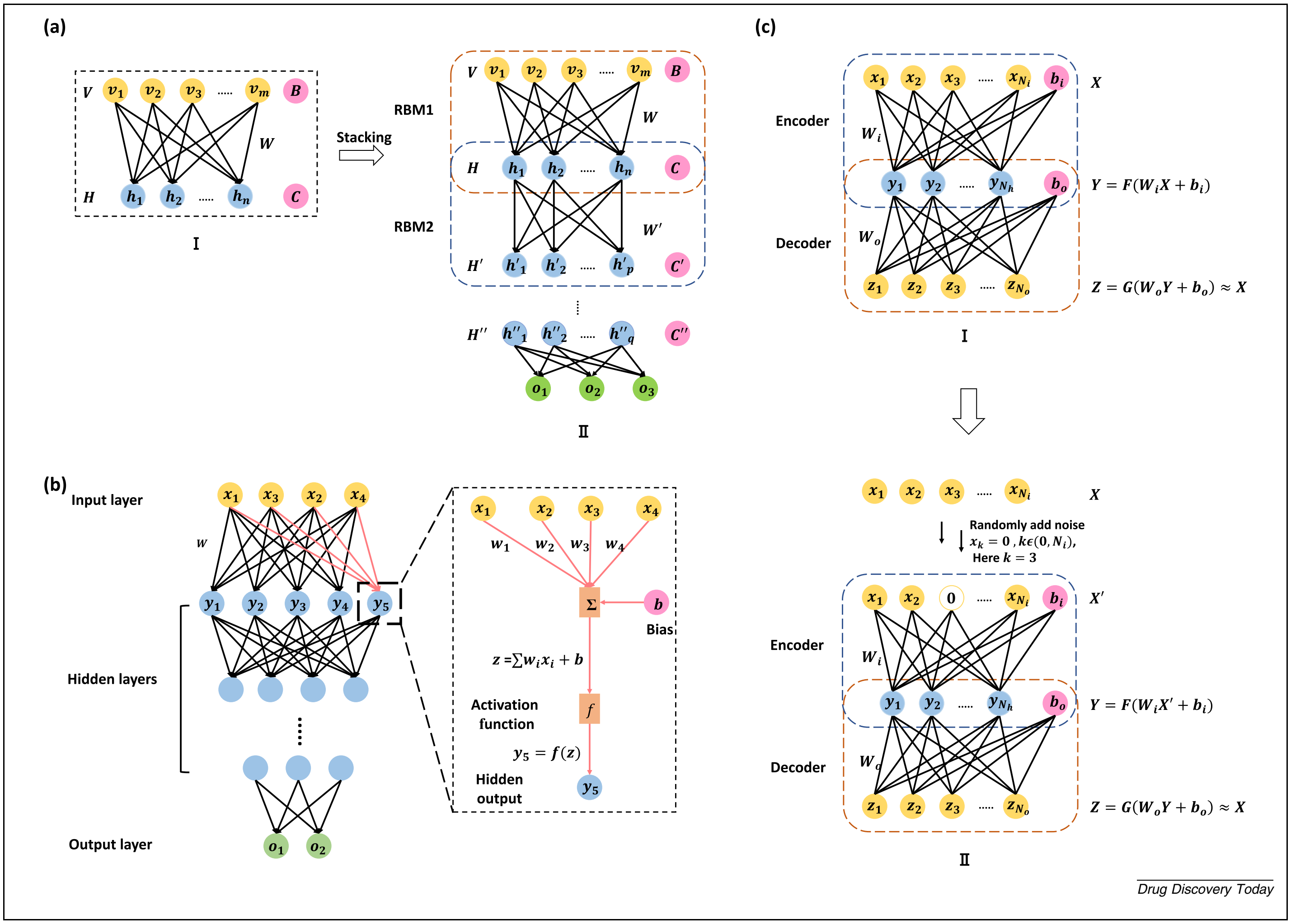


FIGURE 1

The basic framework of (a) deep belief networks (DBN), (b) deep neural networks (DNNs), and (c) AutoEncoders (AEs).

and weaknesses. Finally, we provide possible ways of addressing current challenges to aid further efforts to exploit the full potential of DL in drug repositioning.

Overview of DL frameworks

A common DL architecture comprises a sequence of layers with nonlinear processing units, each layer of which is trained on an ensemble of key features from the output of the previous layer. Herein we summarize six types of commonly used DL framework: deep belief networks (DBNs), DNNs, autoencoders (AEs), convolutional NNs (CNNs), recurrent NNs (RNNs), and graph NNs (GNNs).

Deep belief networks

DBNs are generative NNs commonly used in image recognition and generation (Fig. 1a). Generally, they comprise stacked restricted Boltzmann machines (RBMs),[32] which are structured generative stochastic frameworks with two layers: a visible layer and a hidden layer. The visible units $V = (v_1, v_2, v_3, ..., v_m)$, corresponding to the input data, are connected to the hidden units $H = (h_1, h_2, h_3, ..., h_n)$. All states of the units are binary (i.e., 0 or 1). An energy $E(V,H)$ is calculated to estimate the probability of the input using Eq. (1):

$$E(V,H) = -WVH - BV - CH,$$

where W denotes the weight between two layers, V and H represent the states of the visible units and hidden units, respectively, and B and C are the biases. The margin probability distribution $P(V)$ is calculated using Equation (2):

$$\begin{aligned} P(V) &= \textstyle\sum_H \frac{e^{-E(V,H)}}{S} \\ S &= \textstyle\sum_V \sum_H e^{-E(V,H)} \end{aligned} \tag{2}$$

where S is a normalization function to ensure that the probability value is in the range of $(0, 1)$. The objective of the RBM is to search for probability distribution $P(V)$ that fits best with the input data V so that the network can minimize the discrepancy between the input and generated values.

Given that a single RBM might not fully represent the complex information with binary units, DBNs were developed to treat the hidden layer from the previous RBM as the visible layer of the subsequent RBM. In such a way, the training parameters can grow sharply. To relieve pressure on training cost and improve the performance of DBNs, Hinton et al. proposed an unsupervised, layer-wise, greedy learning algorithm to search for a proper set of initial parameters. This fine-tuning algorithm is further used to optimize the parameters and yield the final predictive model.[33]

Deep neural networks

DNNs are multilayer perceptron (MLP) networks, also known as multilayer fully connected NNs.[34] They use several neurons to learn high-level features from the data by emulating the way in which the human brain works. The typical structure of DNN comprises an input layer, more than two hidden layers, and an output layer (Fig. 1b); each layer is connected to its following layer, which comprises several perceptrons, namely neurons in the NN. Once each neuron receives several input vectors, it performs a dot-product computation between the input and corresponding weights and then adds the bias term to produce the weighted sum. Then, the weighted sum is fed into an activation function, such as the sigmoid and rectified linear unit (ReLU), to perform nonlinear transformation, which eventually gives the final output of this neuron (Fig. 1b).

The purpose of DNN training is to obtain optimal values of the learnable parameters (e.g., weights and bias terms) by using large, labeled data sets. During the training process, the parameters can be adjusted based on the backpropagation (BP) algorithm to minimize errors between the real labels and the predicted results. Compared with traditional ML methods, DNNs can capture high-level, abstract, and complex features from original data without data preprocessing and feature extraction, and show improved prediction performance. Usually, DNNs containing more hidden layers could capture more essential features from input data. However, with the increasing number of hidden layers, there are hundreds of millions or even billions of trainable parameters, which increases the computation cost and possibly cause gradient vanishing issues. Another limitation of DNNs is that they require numerous data for model training to avoid overfitting and local optimality; thus, they are not suitable for small-sample learning.

Autoencoders

Originally, AEs were developed from a one-hidden-layer MLP and utilized to reduce the dimension of the feature space.[35] A basic AE includes an input layer, a hidden layer and an output layer, which conceptually form the encoder and decoder (Fig. 1c). First, an AE takes $X = (x_1, x_2, x_3, ..., x_{N_i})$ as input and exploits the encoder to map the input vector into the low-dimension representation $Y = (y_1, y_2, y_3, ..., y_{N_h})$. Then, the decoder reconstructs the compressed feature Y to obtain the output $Z = (z_1, z_2, z_3, ..., z_{N_o})$, the dimension of which equals that of the initial input X. N_i, N_o, and N_h are the unit numbers of each layer respectively and $N_i = N_o > N_h$.

$$Y = F(W_iX + b_i) \tag{3}$$

$$Z = G(W_oY + b_o) \tag{4}$$

The computation process for the encoder and decoder is described by Eqs. (3) and (4), respectively, in which F and G represent nonlinear activation functions, W_i and W_o denote the weight matrices, b_i and b_o are bias vectors. During the learning process, all parameters (e.g., W_i and b_i) are optimized to minimize the reconstruction error e by a BP algorithm, in which the classical loss function square error L is often adopted (Eq. (5)).

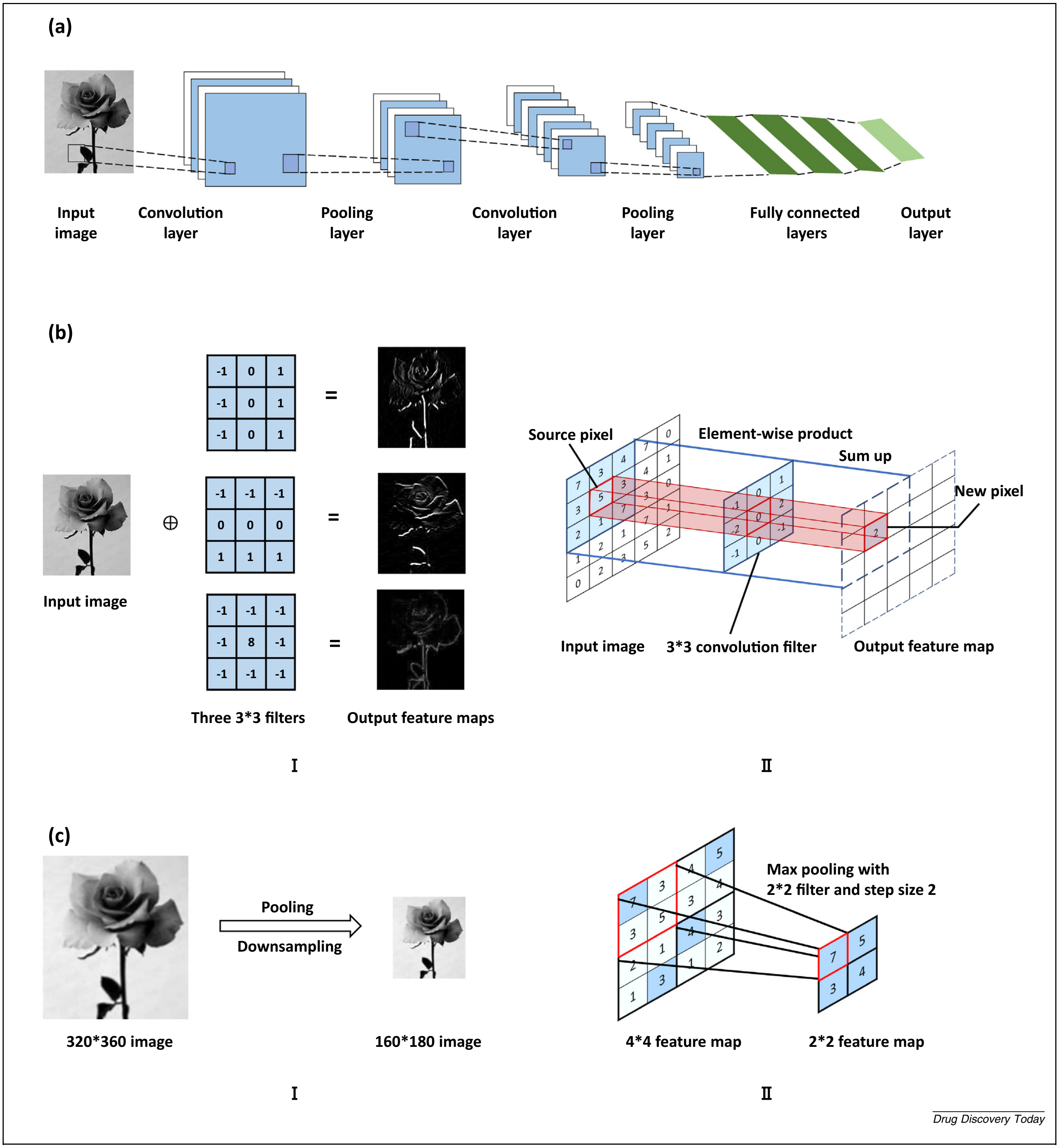
$$\begin{aligned} e &= \tfrac{1}{N_i}\textstyle\sum_{j=1}^{N_i} L(x_j, z_j) \\ L(x, z) &= ||x - z||^2 \end{aligned} \tag{5}$$

However, when it comes to complex inputs, basic AEs might be unable to encode the information from the inputs. As a result, several variants of AE with different characteristics have been developed, such as denoising AE (DAE),[36] variational AE (VAE), and sparse AE (SAE); of these, DAE is relatively more popular. Compared with basic AEs, DAE has a special layer inserted between the input layer and hidden layer (Fig. 1c). The additional layer is used to generate noise for the input X via sparsity limitation or a dropout scheme (i.e., randomly changing input values to 0). Next, the disturbed input $X' = (x_1, x_2, x_3, ..., 0, ..., x_{N_i})$ goes through the same encoding and decoding steps of a basic AE to give the output Z. Importantly, DAE is able to rebuild the original input after training by minimizing the difference between Z and X instead of X'.

Convolutional neural networks

CNNs are feedforward NNs inspired by animal visual cortex research.[37] They use a set of convolution filters to simulate the response of visual cells and have led to remarkable progress in the computer vision field. A basic CNN comprises a series of convolution layers, pooling layers, and several fully connected layers with images as input (Fig. 2a). The convolution layer, which is the key part of CNN, is used to extract various feature maps of the input image by using convolutional filters. Applying several different convolutional filters in a convolution layer can generate different feature maps, which represent various kinds of characteristic of the input image, such as vertical features, horizontal features, and edge features. Fig. 2b illustrates how to yield a feature map by using the convolution operation. When the filter is moved at each location of the input image, an element-wise product between each source pixel and each element of the filter is performed; the results are then summed up to yield the new pixel value in the corresponding position of the output feature map. Generally, to enhance the nonlinear ability of CNN, the activation function is introduced to transform the output from the convolutional layer. There are two main features in the convolutional layer[37]: the weights are shared in the same feature map and the local connectivity is preserved by learning correlations among neighboring pixels. Both of these help reduce the number of learnable parameters and achieve the translation invariance of the input image.

Subsequently, the pooling layer is set to achieve spatial invariance by reducing the dimensionality of feature maps and the number of parameters (Fig. 2c).[38] For a pooling layer, there are no trainable parameters and the number of output feature maps remains unchanged. In particular, max pooling and average pooling are the most common strategies for subsampling.[39] For example, after applying a max-pooling operation, a feature map with the size of $4 * 4$ can be downsampled into the new feature map with a size of $2 * 2$ by selecting the maximum value over the $2 * 2$ pooling window (Fig. 2c).



-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

-1	-1	-1
-1	8	-1
-1	-1	-1

Drug Discovery Today

FIGURE 2

The convolutional neural network (CNN) architecture. (a) The basic CNN framework. (b) Examples of three feature maps developed by using three different convolution filters, where ⊕ represents the convolution operation; the three extracted feature maps contain vertical features, horizontal features, and edge features, respectively. (c) Convolution operation to generate a feature map with a filter size of 3*3: when the filter is moved at each location of the input image, an element-wise product between each source pixel and each element of the filter is performed; these are then summed to yield the new pixel value in the corresponding position of the output feature map.

The final module in a CNN is a multilayer fully connected NN, in which each neuron in current layer is connected to all the outputs of the previous layer. It is responsible for performing high-level reasoning on these output feature maps. Notably the feature maps need to convert into the feature vector before entering fully connected layers.

Recurrent neural networks

RNNs are artificial NNs for handling sequential data by introducing internal memory units,[40] and have been used successfully in natural language processing (NLP). The basic RNN model includes the input layer, the hidden layer and the output layer (Fig. 3a). It receives a sequence $X = (x_1, x_2, \cdots, x_{t-1}, x_t, x_{t+1}, \cdots)$ as the input, where x_t is the word vector at step t, and then outputs the corresponding sequence $O = (o_1, o_2, \cdots, o_{t-1}, o_t, o_{t+1}, \cdots)$ by using the internal units. The output vector o_t at time step t is computed using Eq. (6):

$$h_t = f(Ux_t + Wh_{t-1}) \tag{6}$$

$$O_t = g(Vh_t)$$

where U, W, and V are the weight matrices, f and g denote nonlinear activation functions, and h_t represents the hidden state of the step t, also known as the memorial unit. In theory, h_t captures all the learned features from the forward sequence by involving the current input x_t and previous hidden state h_{t-1}. Different from other regular DL models, the output O_t at the step t is not only dependent on current input (x_t), but also correlated with the prior information (h_{t-1}). Therefore, RNN is good at processing sequential data, such as text and time series data.

When learning the long-term dependency of long sentences, the basic RNN suffers from gradient-vanishing and explosion problems. To overcome these limitations, several RNN variants have been developed, such as long-short-term memory (LSTM)[41] and the gated recurrent unit (GRU).[42] LSTM uses three gates (the input gate, output gate and forget gate) to decide what to keep in memory and what to discard; and it adds the memory unit to store the information before passing to the next unit. This effectively addresses the issue of gradient vanishing when capturing long-term dependencies. Different from LSTM, GRU solves the long-term dependency problem by adopting the reset gate and the update gate to control the flow of information entering the unit. In addition, bidirectional RNNs (Bi-RNNs),[43] such as bidirectional LSTM (Bi-LSTM),[44] are proposed to capture the feature dependency of sequence data. In forward and backward directions, the concatenation of the forward and backward hidden states $[\overrightarrow{h_t}, \overleftarrow{h_t}]$ at step t is fed into the next hidden unit to update h_{t+1}. The current state of Bi-RNN not only contains long-term dependencies from previous time steps, but also considers the future information of sequences. Bi-RNN has been demonstrated to have better prediction accuracy compared with basic RNNs.[43,45]

Graph neural networks

To solve the problem of loss of topological information for graph-structure data by NNs, GNN was pioneered by Gori and Scarselli as an effective tool to directly process graph-structure data.[46–47] Currently, GNN has been widely used in various fields, such as chemistry, natural language, and social networks. In the architecture of GNN, a graph G is treated as a pair of nodes set N and edges set E, where nodes denote the objects or concepts in the graph representation and edges denote their relationships (Fig. 3b). Each node n is connected to its neighbor nodes (ne[n]) by edges and assigned with a label l_n containing feature information. In such a network, a state vector s_n is used to describe the state of the instance corresponding to each node n (Fig. 3b). It is intuitively influenced by the neighbor nodes ne[n] and mathematically represented as the solution of Eq. (7) (for nonpositional graphs),

$$s_n = \sum_{u \in ne[n]} f_w(l_n, s_u, l_u), \quad n \in N \tag{7}$$

where w is a set of parameters and the transition function f_w is used to explain the impacts of the neighbor nodes. Given the states and the label information, the output o_n could be denoted as Eq. (8):

$$o_n = g_w(s_n, l_n), \quad n \in N \tag{8}$$

where g_w is the output function. Originally, the transition function f_w and output function g_w were implemented by feedforward NNs (FNNs).

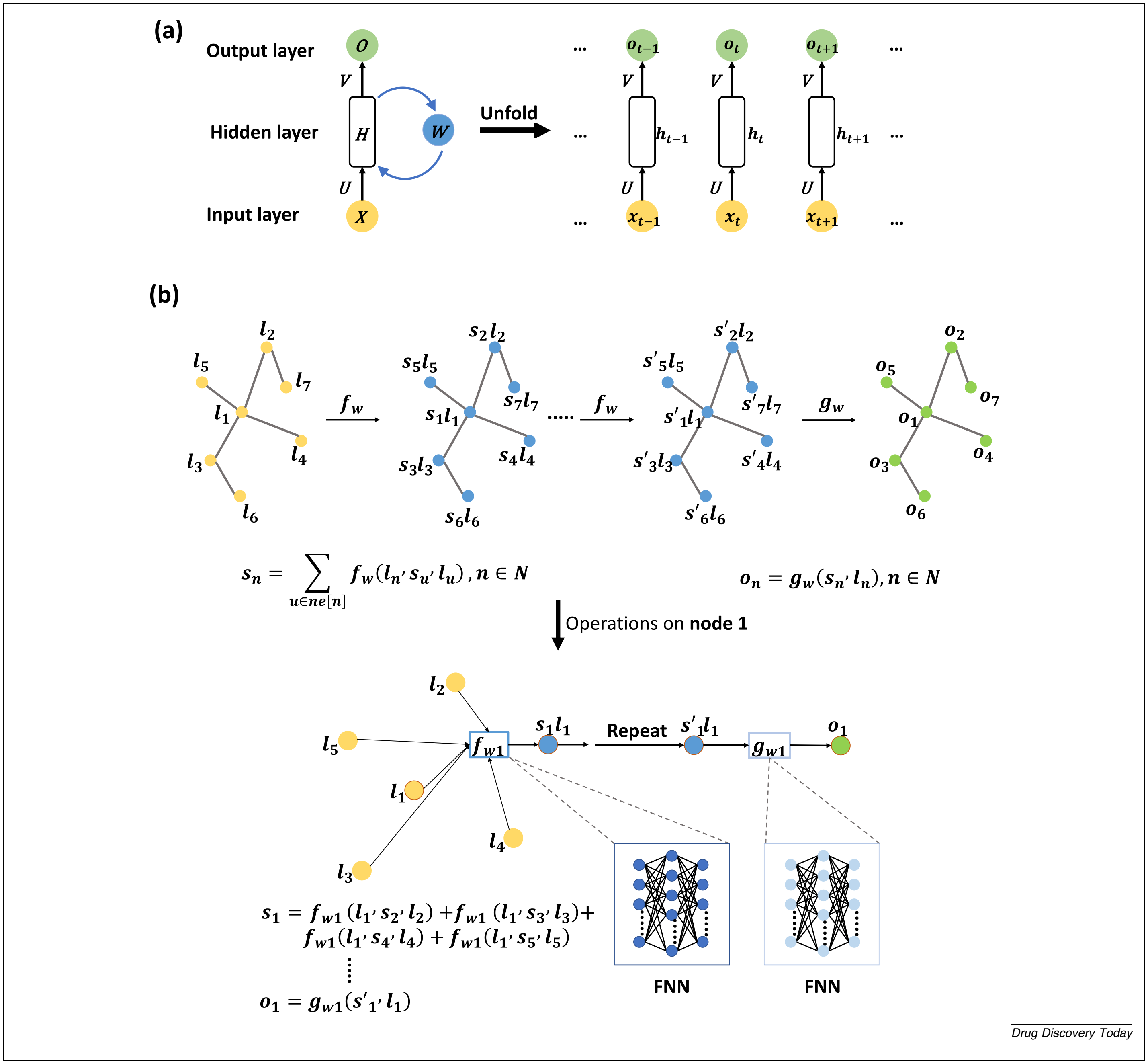
The above-described DL architectures have attracted attention for their potential in target prediction and drug repositioning. They have the advantage of dealing with chemical space or feature space, and could capture connotative representations in DTIs/drug–target associations, compared with traditional computer-aided drug design methods, such as molecular docking and pharmacophore modelling. However, how to select the appropriate DL architecture to treat different data forms and target problems remains an open question. The following summaries and discussions of recently reported DL approaches for drug repositioning will be useful to promote thinking of this state-of-the-art technology in drug discovery.

Drug–target interaction-based DL approaches

DTIs are direct, relatively uniform pieces of information that can be used for establishing predictive models to identify new DTIs and eventually achieve drug discovery and drug repositioning tasks. Most previous ML methods (e.g., energy-component, empirical, and knowledge-based scoring functions)[48] were established to predict DTIs by using explicit rules to describe the protein–ligand interaction modes. In recent years, several different DL approaches have been developed for DTI prediction by learning high-level implicit features from various engineered data. These DL approaches are summarized and compared according to the two types of DTI data form: complex-based inputs and noncomplex-based inputs.

Complex-based inputs

Complex-based inputs refer to those directly encoded from protein–ligand complex structures, such as feature-embedded 3D grid representations and graph representations. DL NNs are usually established to learn from complex-based inputs to predict protein–ligand-binding affinity. Combined with molecular docking and molecular dynamics methods, the established DL models could work as scoring functions to reassess the predicted binding


(a)
Output layer
Hidden layer
Input layer
O
V
H
W
U
X
Unfold
o_{t-1}
o_t
o_{t+1}
h_{t-1}
h_t
h_{t+1}
x_{t-1}
x_t
x_{t+1}
(b)
l_1 l_2 l_3 l_4 l_5 l_6 l_7
f_w
s_1l_1 s_2l_2 s_3l_3 s_4l_4 s_5l_5 s_6l_6 s_7l_7
f_w
s'_1l_1 s'_2l_2 s'_3l_3 s'_4l_4 s'_5l_5 s'_6l_6 s'_7l_7
g_w
o_1 o_2 o_3 o_4 o_5 o_6 o_7
$s_n = \sum_{u \in ne[n]} f_w(l_n, s_u, l_u), n \in N$
$o_n = g_w(s_n, l_n), n \in N$
Operations on **node 1**
f_{w1}
s_1l_1
Repeat
s'_1l_1
g_{w1}
o_1
$s_1 = f_{w1}(l_1, s_2, l_2) + f_{w1}(l_1, s_3, l_3) + f_{w1}(l_1, s_4, l_4) + f_{w1}(l_1, s_5, l_5)$
$o_1 = g_{w1}(s'_1, l_1)$
FNN
FNN

Drug Discovery Today

FIGURE 3

Recurrent neural networks (RNNs) and graph neural networks (GNNs). (a) The basic RNN architecture comprises an input layer, hidden layer, and output layer. X represents the input sequence, H is the hidden state containing all the extracted features from X, and O denotes the predicted output given input X. U, W, and V are the weight matrixes for the transition from X to H, H to H, and H to O, respectively. (b) General computation process of GNNs.

poses and use for drug repositioning (Fig. 4a). This is expected to enhance the predictive ability and efficiency of drug repositioning. The core DL approaches are discussed herein.

Ragoza et al. proposed a 3DCNN scoring model for pose prediction and virtual screening.[51] In their work, they adopted 3D grid representations to describe the protein–ligand complexes. They treated the complex structures as 3D images with several channels that denoted different features according to smina atom types.[52] In a specific channel, each grid stored corresponding atom type information and treated a given atom as a density distribution related to its van der Waals radius. With this form of molecular characterization, the complex features were processed by a basic 3DCNN model to discriminate whether the protein–ligand interaction takes place. This model outperformed conventional docking scoring (e.g., Vina-Score) and ML methods (e.g., RF-Score and NNScore) when tested on CSAR-NRC and DUD-E data sets.

With more efforts focussed on the model construction of 3DCNN, Jiménez et al. developed a similar DL approach, named K_{DEEP}, to predict protein–ligand-binding affinity.[53] They used voxel occupancies to represent various features, including seven atom types and excluded volume,[54] and conducted data augmen-

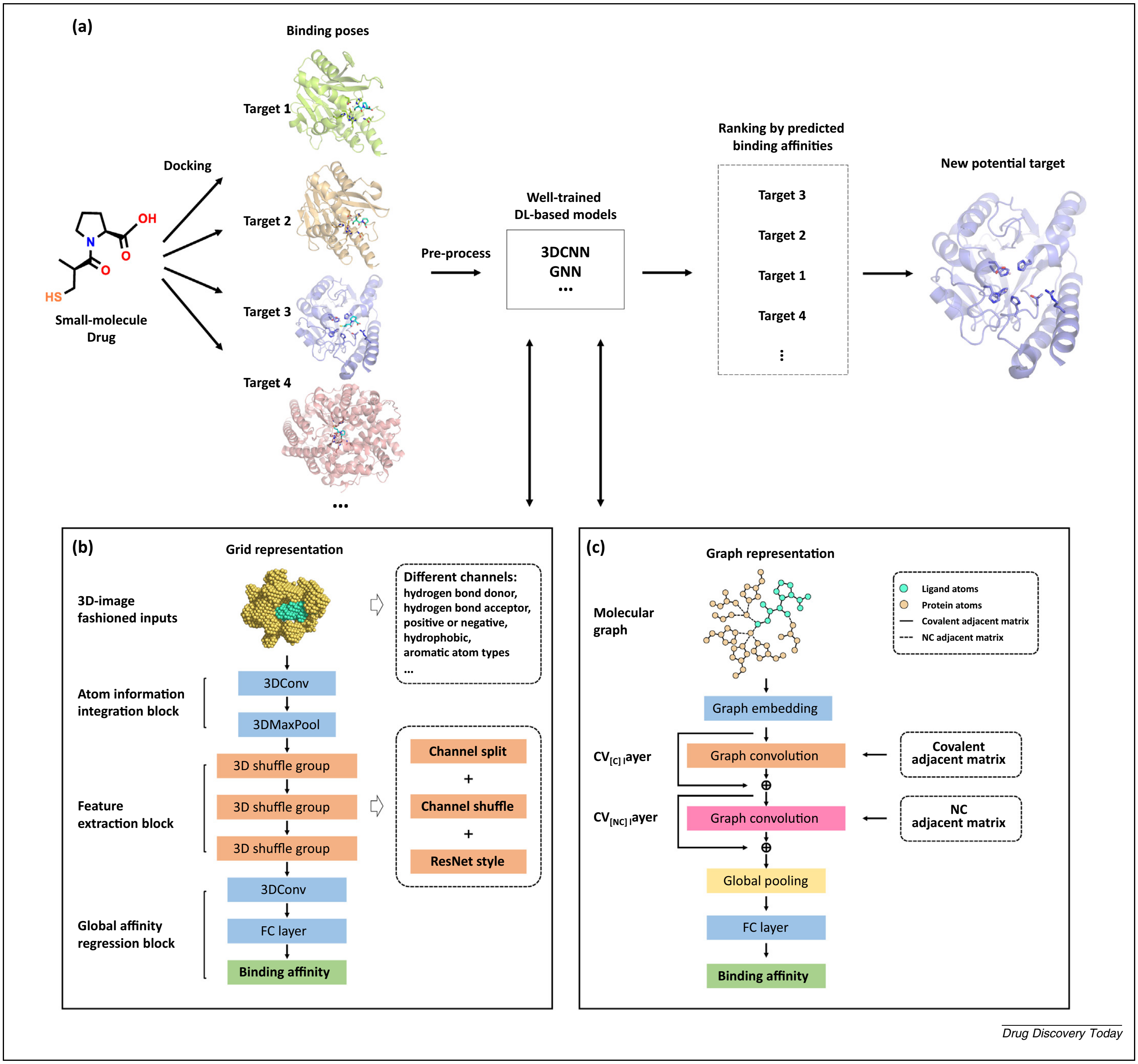


FIGURE 4

(a) General pipeline of drug repositioning with deep learning (DL) frameworks using complex-based inputs. Two representative DL approaches: (b) DeepAtom[49] and (c) InteractionNet.[50]

tation by rotating the complex to deal with the problem related to position sensitivity. Moreover, they improved model construction by introducing SqueezeNet,[55] which uses fewer parameters but has comparable performance to some typical networks (e.g., AlexNet[56]when applied for image recognition. The testing results indicated that K_{DEEP} achieved better predictive ability than many traditional scoring functions (e.g., RF-Score and X-Score).

Inspired by light-weight networks (e.g., Xception,[57] MolbileNet,[58,59] and ShuffleNet[60,61]), Rezaei et al. built a new architecture named DeepAtom to balance the performance and complexity of the model (Fig. 4b).[49] The core part of DeepAtom lies in the 3D shuffle group, which comprises stacked 3D convolutional layers with small kernel sizes and strides; this enables a decrease in the parameter number but increased model depth and complexity. In addition, the authors adopted occupancy descriptors[54] and further considered pharmacophore-like Arpeggio atom types.[62] By testing on PDBbind, DeepAtom outperformed the predictive ability of other traditional scoring functions and related DL methods (e.g., Pafncy). The authors also highlighted that DeepAtom has the ability to generalize protein–ligand interaction information by the use of moderate-level atom features, suggesting that accurate representation of protein–ligand interactions will be helpful to improve model performance.

Different than those approaches mentioned above, Feinberg et al. used basic information of atoms, bonds and distances, instead of other complex knowledge-based features. They proposed PotentialNet,[63] a GNN network for protein–ligand affinity and molecule property prediction. Motivated by the distance matrix, they developed a novel adjacency matrix in the form of a tensor with shape [N, N, N_{et}], where N denotes the total atom number in the complex and N_{et} the number of edge types, so that the matrix is able to include both covalent bond information and noncovalent interactions. PotentialNet is a spatial graph convolutional network with GRU (frequently used in gated GNNs[64]) as the update function. It first took two steps to update the node features, including covalent-only propagation and noncovalent and covalent propagation. Then, the ligand graph went through the graph gather layer and FC layers to achieve the predictive affinity or molecular properties. PotentialNet was observed to have stronger predictive ability than RF-Score and X-Score when used for binding affinity prediction. Meanwhile, variants of PotentialNet were also demonstrated to be state-of-the-art methods for the prediction of molecule properties.

Similarly, Lim et al. constructed a GNN-based framework to predict DTIs,[65] which directly extracted the spatial information from the binding poses. To represent the structural information, they adopted two adjacent matrices, including $A1$ (denoting purely covalent interactions) and $A2$ (denoting covalent interactions and noncovalent intermolecular interactions), and node feature x for protein and ligand. The model dealt with the node feature x in two different ways according to $A1$ and $A2$ to give two continually updated node features, $x1$ and $x2$, respectively; the difference between the monomers and the complex was represented as the discrepancy between $x1$ and $x2$. Most importantly, during the whole training process, a distance-aware graph attention mechanism was applied to distinguish the contribution of different neighbor nodes to a given node and put more emphasis on the residues with less distance from the ligand. Additionally, to enhance the performance of the model, the gate mechanism was used to embed the node features from the previous layer. When tested on the DUD-E and PDBbind data sets, this method outperformed several previously reported DL models.

Furthermore, Cho et al. proposed another GNN architecture, InteractionNet,[50] to predict the dissociation constants between ligands and proteins. Enlightened by PotentialNet, the covalent and noncovalent adjacency matrices were used to represent the protein–ligand complex. Distinct from PotentialNet, they used two consecutive graph convolution layers [i.e., covalent convolution ($\text{CV}_{[\text{C}]}$) and noncovalent convolution layers ($\text{CV}_{[\text{NC}]}$)] to update the node features according to covalent and noncovalent adjacency matrices, respectively (Fig. 4c). In addition, a global pooling layer was exploited to integrate the node features instead of the graph gather layer (in PotentialNet). Based on the refined set of the PDBbind v2018 data set, the authors performed 20-fold cross-validated experiments to train the model and verify the significance of the $\text{CV}_{[\text{NC}]}$ layer. The well-trained model showed better performance than PotentialNet. The authors also tried layer-wise relevance propagation (LRP) to assess the prediction results in human terms with knowledge-based analysis. Notably, with the LRP and visualization process,[66] InteractionNet successfully distinguished the functional groups from the input data and identified actual hydrogen-bond interactions between protein and ligand, implying its explanation ability and reliability. Owing to the superior protein–ligand-binding affinity prediction, we anticipate more applications of these DL approaches in drug discovery and repositioning.

Noncomplex-based inputs

Noncomplex-based inputs refer to those from separated target information and ligand/drug information. Therefore, in contrast to complex-based inputs, this input manner does not require protein–ligand complex structures or atomic-level interaction models. To handle this kind of input, current NNs are designed to learn the features of proteins and ligands in parallel, and, most commonly, the two sets of learned features are concatenated and then passed to the FC layers to learn their interaction information and to give a result according to the task-specific label values. In other words, given the ligand and target information, the trained DL models will predict whether the ligand has interactions with the target.

For example, Wen et al. adapted a DBN architecture called DeepDTIs to predict protein–ligand binding,[70] in which ECFP was used to encode ligand and sequence composition descriptors was used to represent protein. Based on the DrugBank and EDTPs data sets, DeepDTIs involved a two-step training process: greedy layer-wise unsupervised training and supervised fine-tuning methods. Consequently, DeepDTIs showed a good predictive ability, which outperformed traditional ML methods, including Bernoulli Naive Bayesian, Decision Trees (DT), and RF in the used test set.

Öztürk et al. reported DeepDTA,[67] a CNN framework, to predict drug–target-binding affinity by using separated amino acid sequence information of targets and SMILES information of drugs. DeepDTA comprises two separate CNN blocks to learn representations from protein sequences and drug SMILES strings, respectively (Fig. 5a). Three consecutive 1D-convolutional layers followed by a max-pooling layer were used to obtain advanced representations of targets and drugs, which then were concatenated and fed into three core FC layers (Fig. 5a). The activation function ReLU and the loss function MSE were applied to minimize the difference between the experimental binding affinity values and the prediction values. The established models were evaluated on two benchmark data sets: Kinase data set Davis and KIBA data set, which showed superior performance in drug–target-binding affinity, outperforming the KronRLS and SimBoost algorithms.

Similarly, Karimi et al. developed a unified RNN-CNN network,[71] named DeepAffinity, to solve the regression problem of binding affinity prediction. They used pretrained RNNs for data initialization and appended the CNN model comprising a 1D convolution layer and a max-pooling layer to abstract high-level features. The outputs of the CNNs for targets and compounds were concatenated and fed into two fully connected layers. Pretrained RNN initializations were demonstrated to have important roles in the non-convex training process. Notably,

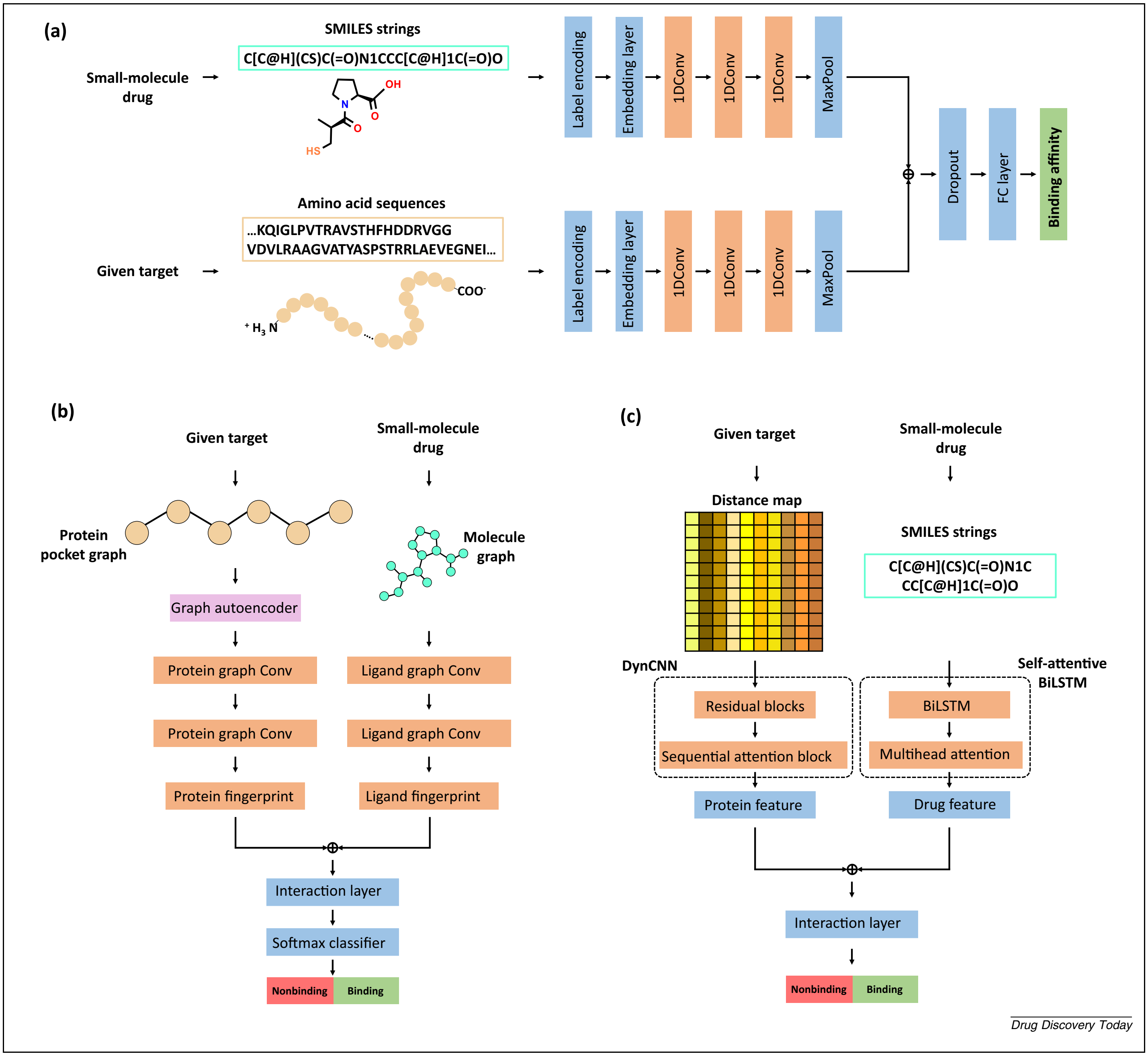


FIGURE 5

Representative deep learning (DL) frameworks using noncomplex-based inputs: (a) DeepDTA,[67] (b) graph convolutional neural network (CNN),[68] and (c) DrugVQA.[69]

the authors embedded attention mechanisms in the RNN-CNN network to enhance the model interpretability. This model showed better performance in binding affinity prediction compared with RF and the separated RNN and CNN models, and also had good performance in target selectivity prediction, suggesting its potential in drug repositioning.

Torng et al. constructed a binary-classified Graph CNN framework with unsupervised pocket graph AE to predict DTIs.[68] Different from the regular GNN architectures described above, the authors used the pocket graph and ligand graph separately as inputs. The pocket graph AE was pretrained to learn essential embeddings of protein pocket, followed by using the graph convolutional layers to further extract key information for protein and ligand graphs (Fig. 5b). The extracted features of protein and ligand were joined to an interaction layer to learn their interaction information. The testing results indicated that Graph CNN had better performance than 3DCNN, Vina, RF-Score, and NNscore.

Recently, to consider 3D spatial information, Zheng et al. proposed a novel network named DrugVQA using a quasi-visual question answering system to predict drug–protein interactions.[69] Uniquely, 2D pairwise distance maps generated from 3D protein structures were adopted to balance the lack of 3D spatial information and low efficacy. Dynamic attentive CNN was also designed to process the protein features and determine the importance of each amino acid (Fig. 5c). For drug molecules, the authors used self-attentive BiLSTM to encode the SMILES strings into 2D embedding matrices (Fig. 5c). The protein and ligand features were concatenated and fed into the classification layer to get a result. By testing on different benchmark sets, DrugVQA had superior performance compared with ML scoring functions (e.g., NNscore and RF-score), docking-based methods (e.g., Vina) and DL-based methods (e.g., 3D-CNN, AtomNet, PocketGCN, and GNN). This work revealed the remarkable capability of the distance map representation for use with protein targets and the robustness of the unique model.

In addition, Stokes et al. established a D-MPNN-based model by building a molecular representation based on a specific property (i.e., inhibition of the growth of *Escherichia coli*) using a directed message-passing approach.[28] They trained the model using 2335 diverse molecules that inhibited the growth of *E. coli*, and augmented the model with several molecular features, hyperparameter optimization, and ensembling. They finally used the model against multiple chemical libraries to identify potential lead compounds with activity against *E. coli*. According to the predicted score and experimental validation of the model, the authors successfully identified Halicin, a c-Jun N-terminal kinase inhibitor, as a new broad-spectrum antibacterial compound. This work highlights the potential of DL methods in drug repositioning.

The above-mentioned examples exemplify the importance of developing specific DL architectures together with sophisticated feature engineering skills for DTI prediction tasks. Furthermore, efficient models, such as Graph CNN and DrugVQA, emphasize the accurate and effective description of DTIs, which is likely to reduce computation time and improve model interpretability, and will be appropriate for further investigations.

Heterogeneous network-based DL approaches

In addition to the above-described data forms of DTIs, there are several pharmacological, preclinical, or clinical data forms, such as side effects, phenotypic and genotypic attributes, electronic health records (EHRs), patient surveys, clinical trial data, and postmarketing surveillance data, which constitute complicate heterogeneous networks, and can also be used to establish predictive models using DL approaches for drug repositioning. This is mainly based on similarity hypothesis, that is, that similar drugs have similar target profiles or similar phenotypes (e.g., diseases or side effects). In such graph-structured heterogeneous networks, nodes are usually represented by drugs, diseases, targets, genes, side effects, and others, and edges are denoted as the relationships between two nodes (e.g., chemical similarities between two drugs, interactions between drug and target, or underlying mechanisms of a drug to a disease). Various DL approaches have been established based on these heterogeneous data for drug repositioning. Despite the diversified data sources, these approaches are generally used to predict drug–target (gene) and drug–disease associations, as discussed below.

Predicting new drug–target (gene) associations

Several DL approaches have been established by integrating drug–target associations, drug–drug interactions, target–target interactions, chemical similarities, and/or target sequence similarities to extract underlying features to discover new drug–target (gene) associations. For instance, Eslami Manoochehri and Nourani proposed a DNN-based framework to predict new drug–target associations based on drug chemical similarity and protein sequence similarity (Fig. 6a).[72] They first constructed a DTI network by exploiting known DTIs, drug chemical similarities, and protein sequence similarities. For each drug–target pair, they then used a subgraph extraction algorithm to extract subgraphs from the interaction network, with the aim to represent the topological environment around each drug–target pair. These subgraphs were encoded into embedding vectors and then fed into DNNs to learn the graph topological features to predict unknown DTIs. Compared with four baseline methods (LMNII, CMF, HNM and NetLapRLS), this framework achieved higher performance in terms of area under the receiver operating characteristic (AUROC) and area under the precision-recall curve (AUPRC).

Zong et al. added disease-related information (i.e., drug–disease and target–disease associations) to enrich the drug–target network, generating a drug–target–disease tripartite network.[75] Within such a tripartite network, they identified potential DTIs by adopting a DL algorithm called DeepWalk, which vectorized the nodes (e.g., drugs and targets) based on the local latent topology information and then gave the topological similarity for each drug–drug or target–target pair. Compared with the drug–target bipartite networks, their method showed improved performance, highlighting the importance of the additional disease-related information. Furthermore, they proved that using network topology to compute similarity between nodes outperformed those models based on drug chemical similarity and protein sequence similarity.

To enrich and understand the relationship between drugs and targets more fully, Peng et al. additionally integrated other types of data, such as drug–side-effect associations to characterize drugs and targets, and developed a CNN-based model named DTI-CNN to predict DTIs (Fig. 6b).[73] They constructed four drug-related networks (i.e., drug structural similarity network, drug–drug interaction network, drug–disease association network, and drug–side-effect association network) and three protein-related networks (i.e., protein sequence similarity network, protein–protein interaction network, and protein–disease association network). These similarity networks were represented separately as similarity matrices of drugs and targets, and further encoded into the advanced feature matrices by the random walk

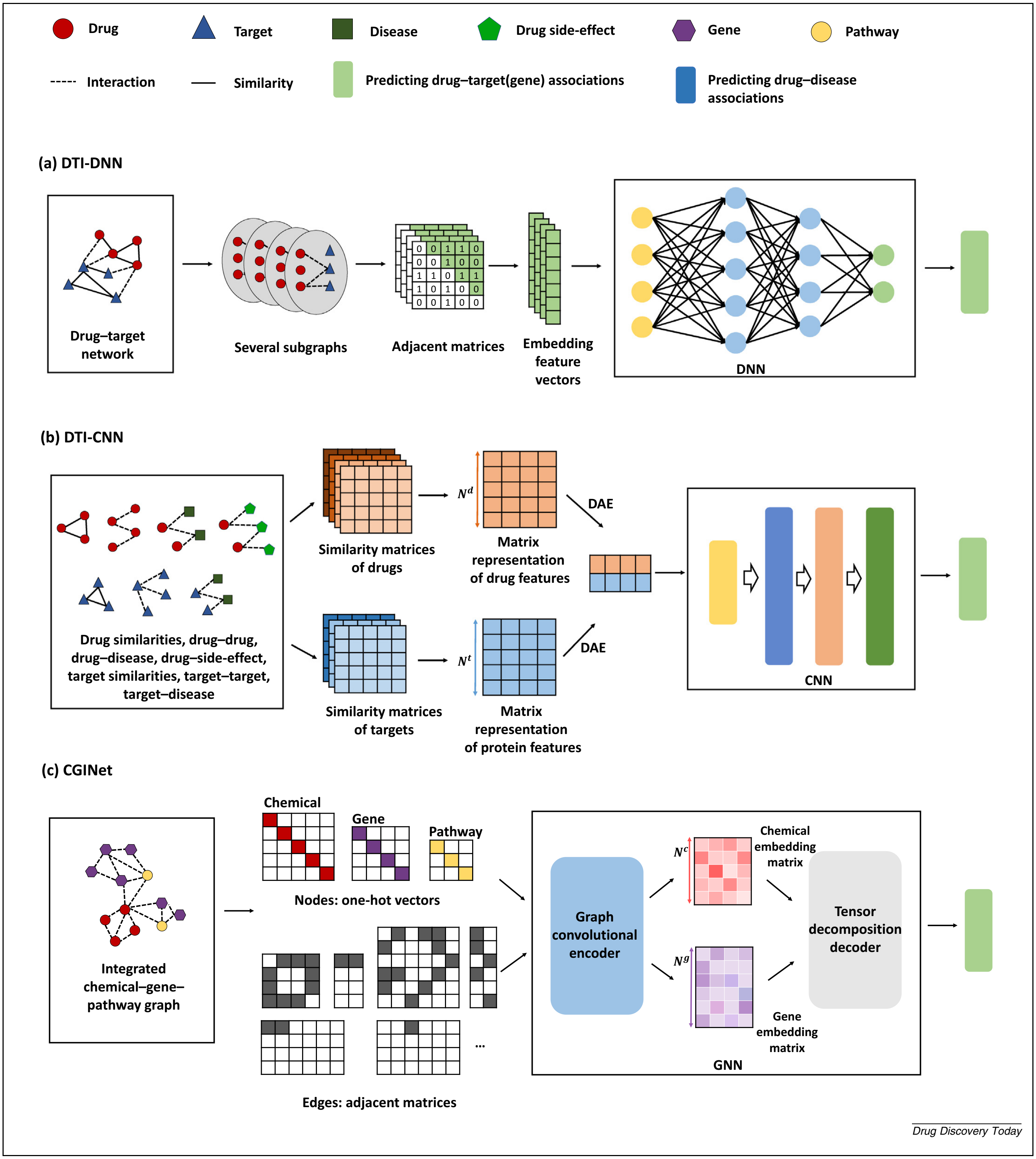

Drug
Target
Disease
Drug side-effect
Gene
Pathway
Interaction
Similarity
Predicting drug–target(gene) associations
Predicting drug–disease associations
(a) DTI-DNN
Drug–target network
Several subgraphs
Adjacent matrices
Embedding feature vectors
DNN
(b) DTI-CNN
Drug similarities, drug–drug, drug–disease, drug–side-effect, target similarities, target–target, target–disease
Similarity matrices of drugs
Matrix representation of drug features
Similarity matrices of targets
Matrix representation of protein features
N^d
N^t
DAE
DAE
CNN
(c) CGINet
Integrated chemical–gene–pathway graph
Chemical
Gene
Pathway
Nodes: one-hot vectors
Edges: adjacent matrices
...
Graph convolutional encoder
N^c
Chemical embedding matrix
N^g
Gene embedding matrix
Tensor decomposition decoder
GNN
Drug Discovery Today

FIGURE 6

Representative deep learning (DL) frameworks for drug–target (gene) association prediction: (a) drug–target interaction (DTI) deep neural network (DTI-DNN),[72] (b) DTI-convolutional neural network (CNN),[73] and (c) CGINet.[74]

model to obtain the initial drug feature and target feature vectors. Notably, DAE was used to reduce the dimensions of drugs and target features to generate their final low-dimensional feature vectors, which were then concatenated and passed to the CNN block to predict the probability of interaction between drugs and targets. The evaluation results showed that DTI-CNN achieved better performance than three traditional methods (DTINet, CMF, and NRLMF), proving that it can effectively learn the topological features and correlations between drugs and targets in the complicated network through layer-by-layer learning.

The interactions between chemicals and genes are also of significance to find novel therapeutic targets for known drugs. Wang et al. proposed a graph convolution network (GCN)-based model, namely CGINet, to predict chemical–gene interactions in an integrated multirelational graph (Fig. 6c).[74] Those subgraphs including chemical–chemical association graphs, gene–gene association graphs, and chemical–pathway association graphs; and gene–pathway association graphs and chemical–gene multi-interaction graphs were used to construct the integrated multirelational graph, in which each node (i.e., chemicals, genes, and pathways) was represented as a one-hot vector, and edges between nodes were denoted by an adjacency matrix. Notably, GCN was trained on these binary association subgraphs and the integrated multirelational graphs to learn more embedding features of chemicals and genes, followed by using the tensor decomposition decoder to compute the interaction probability for a chemical–gene pair. The testing results suggested that the CGINet model exhibited competitive performances compared with the baseline models (DeepWalk, SVD, and Laplacian), and had the ability to predict novel chemical–gene interactions, which did not appear in the original graphs.

Predicting new drug–disease associations

Drug–disease associations provide important, direct information for drug discovery and drug repositioning, whereas such associations are related to several aspects of data, such as disease–gene associations, drug-exposure–gene expression data, and patients' prescription information. On basis of such comprehensive information, several DL approaches were established to search for new drug–disease associations. For example, Liu et al. developed HNet-DNN, a DNN network, to predict new drug–disease associations based on the features extracted from drug–disease heterogeneous networks (Fig. 7a).[76] They collected drug chemical structures and corresponding phenotypic attributes of diseases to build the drug–drug similarity and disease–disease similarity networks, and then combined these with drug–disease associations to establish a drug–disease heterogeneous network. The topological features of drugs and diseases were extracted and combined to train the DNN model for predicting potential drug–disease associations. Compared with the straightforward conjunction of raw features of drugs and diseases as the DNN input, the extracted topological features were markedly condensed and informative, improving the prediction performance. The evaluation results on the PREDICT data set showed that HNet-DNN achieved state-of-the-art performance and outperformed several traditional network-based prediction methods (e.g., LR, SVM, and RF). In case studies, the model successfully predicted some old drugs, such as cisplatin, ethinyl estradiol and carboplatin, for the treatment of breast cancer, which further verified its repositioning ability.

Jarada et al. also proposed a DNN-based framework, named SNF-NN, to predict novel drug–disease associations in the drug–disease heterogeneous network.[77] Uniquely, they used several forms of drug-related information (e.g., chemical structures, target protein sequences, and side effects) and disease-related information (e.g., disease phenotypes, disease–gene associations, and disease–miRNA associations) to construct different similarity networks of drugs and diseases. Before integrating the drug/disease information, they introduced similarity selection and similarity network fusion to reduce data redundancy and inconsistency. Subsequently, highly informative feature vectors of drugs and diseases were obtained from the integrated similarity networks, and then concatenated to feed into the DNN model. SNF-NN was evaluated on three benchmark data sets, which achieved remarkable performance and showed comparable reliability and robustness.

Furthermore, Li et al. introduced a novel DNN-based drug repositioning approach for nonsmall cell lung cancer (NSCLC) by using transcriptomic data and chemical structural information from the MeSH and LINCS databases.[29] This approach included classifying and repurposing processes. In the classifying process, the authors selected 'Landmark genes' and signaling pathways as drug features to feed into DNN to classify each drug into its corresponding therapeutic use category. In the repurposing process, they repurposed the misclassified antineoplastic drugs that belonged to another therapeutic use category as candidates for the treatment of NSCLC. According to the combinational ranking of chemical structure similarity scores and computed pathway activation scores, they successfully repurposed pimozide, an antidyskinesia agent, as a strong candidate to treat NSCLC, which was validated by *in vitro* experiments (Table 1).

Recently, more attempts to use CNN to extract the hidden feature representation of drug–disease associations for drug repositioning have been reported. For instance, Jiang et al. proposed a CNN-based feature extraction method for identifying new drug–disease associations, termed SKCNN (Fig. 7b).[79] They included drug chemical similarity/sigmoid kernel similarity, disease semantic similarity and sigmoid kernel similarity between diseases from the drug–disease association network. Then, this information was fused into one informative feature vector, which comprehensively reflects the characteristics of the disease and drugs. Taking the combined feature vectors of drugs and diseases as input, CNN was trained to learn hidden feature representations of drug–disease associations, and then the traditional RF classifier gave the predicted probability of a drug–disease association. The evaluation results revealed that the prediction performance of SKCNN was superior to the benchmark models MBiRW, Drug-Net, HGBI, KBMF, and DRRs. Several drug candidates predicted by their model for obesity and asthma were further confirmed in the ClinicalTrials.gov database (CTD), demonstrating the efficiency of SKCNN.

Xuan et al. integrated CNN and Bi-LSTM into a novel DL framework named CBPred for discovering new drug–disease asso-

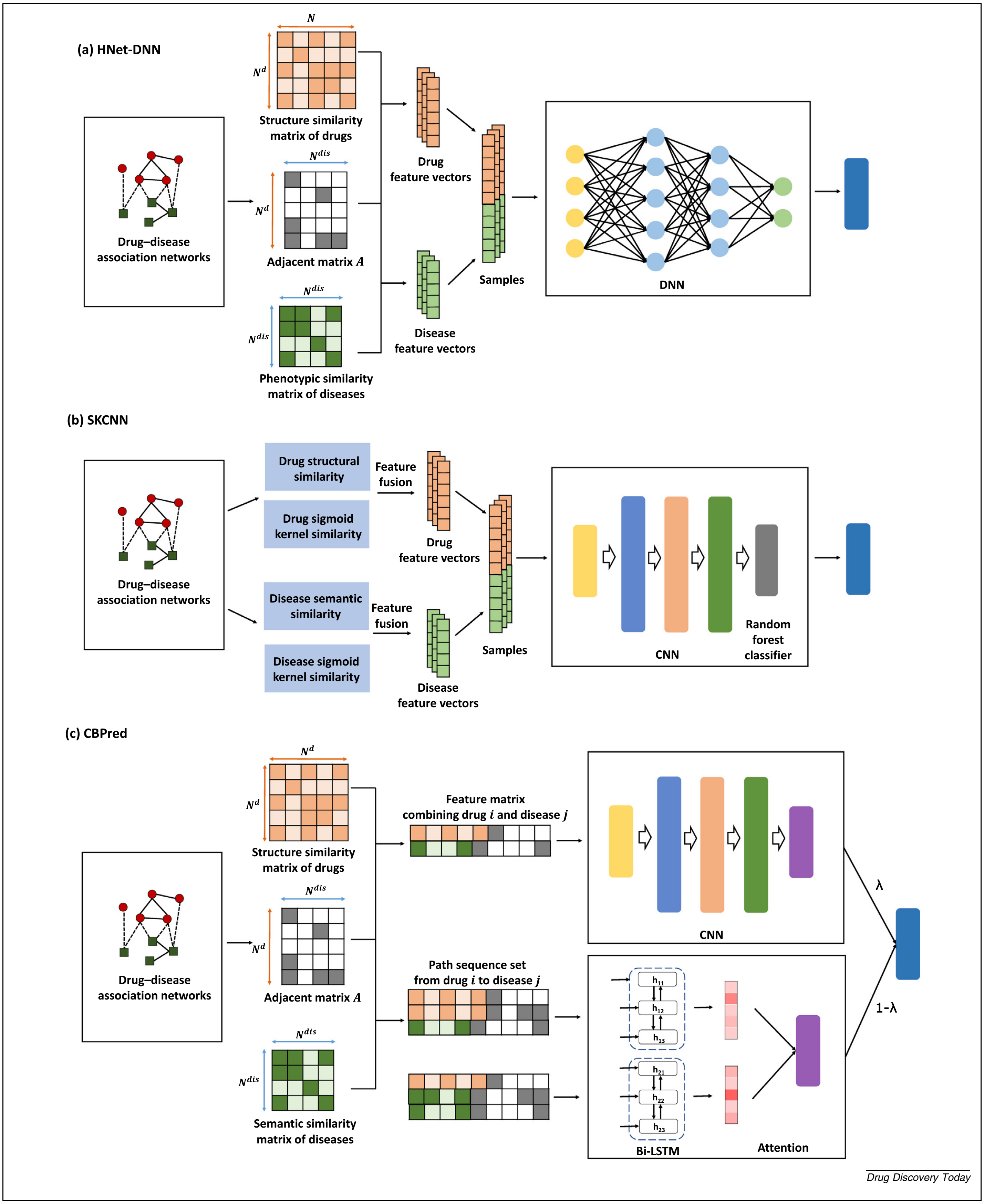

(a) HNet-DNN
N
N^d
Structure similarity matrix of drugs
N^{dis}
N^d
Adjacent matrix A
N^{dis}
N^{dis}
Phenotypic similarity matrix of diseases
Drug–disease association networks
Drug feature vectors
Disease feature vectors
Samples
DNN
(b) SKCNN
Drug–disease association networks
Drug structural similarity
Drug sigmoid kernel similarity
Disease semantic similarity
Disease sigmoid kernel similarity
Feature fusion
Feature fusion
Drug feature vectors
Disease feature vectors
Samples
CNN
Random forest classifier
(c) CBPred
N^d
N^d
Structure similarity matrix of drugs
N^{dis}
N^d
Adjacent matrix A
N^{dis}
N^{dis}
Semantic similarity matrix of diseases
Drug–disease association networks
Feature matrix combining drug i and disease j
Path sequence set from drug i to disease j
CNN
λ
1-λ
h_{11}
h_{12}
h_{13}
h_{21}
h_{22}
h_{23}
Bi-LSTM
Attention
Drug Discovery Today

FIGURE 7

Representative deep learning (DL) frameworks for drug–disease association prediction: (a) HNet-deep neural network (DNN),[76] (b) SKCNN,[77] and (c) CBPred.[78]

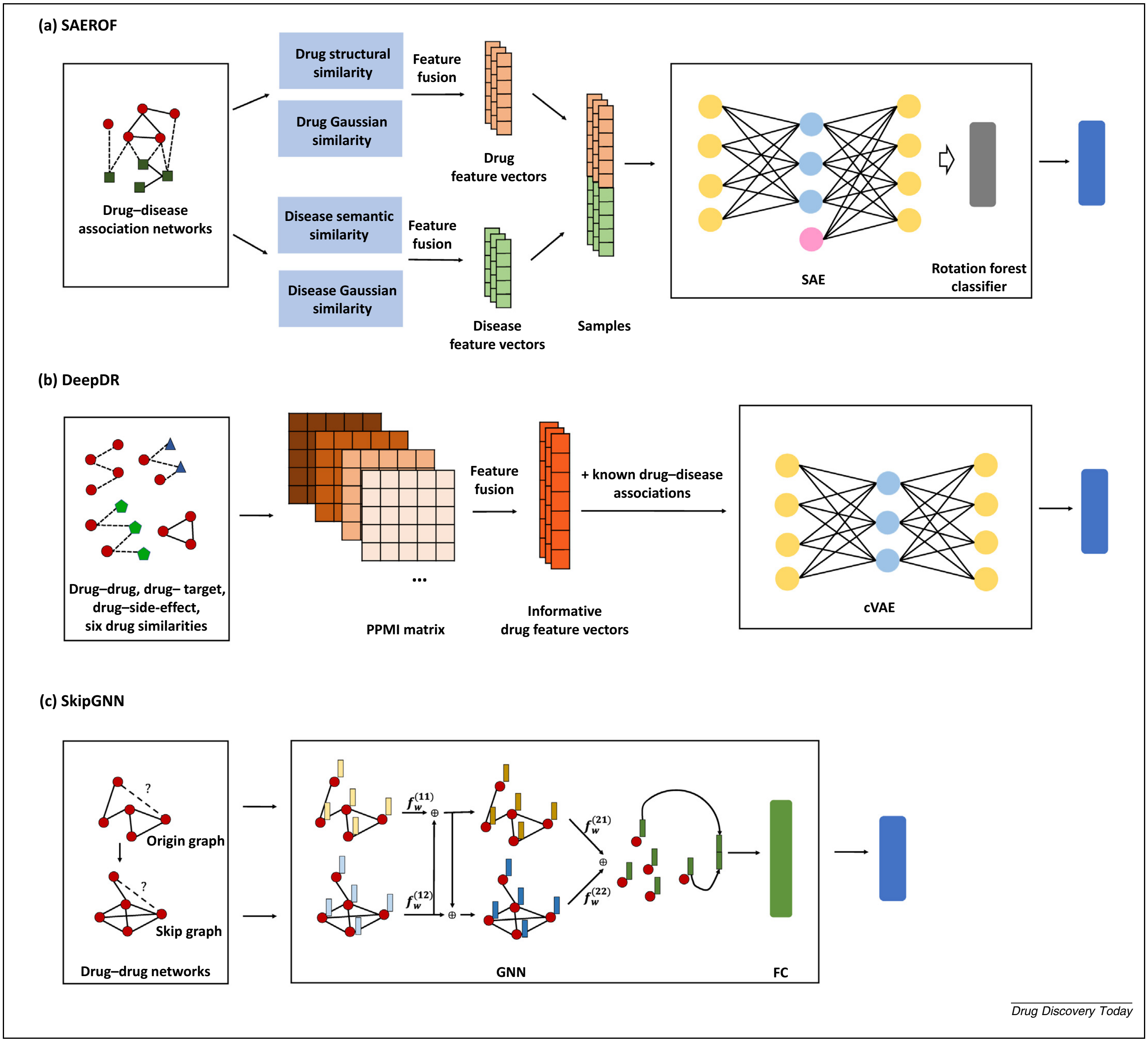


FIGURE 8

Representative deep learning (DL) frameworks for drug–disease association prediction: (a) SAEROF,[80] (b) DeepDR,[81] and (c) SkipGNN.[82]

ciations (Fig. 7c).[78] On the one hand, the CNN module continued to learn the original representation of drug–disease pairs from drug structural similarity, disease semantic similarity, and drug–disease associations. On the other hand, given that the multiple paths between drugs and diseases influence their association possibility, the Bi-LSTM module was established to learn path representations between drugs and diseases by taking the multiple path sequences. Furthermore, the attention mechanism at the path level was introduced to discriminate the different contributions of the path for drug–disease associations. Subsequently, the two high-level feature vectors learned from CNN and Bi-LSTM were sent to the softmax layer, respectively, to get the weighted sum as the final predicted score of a drug–disease association. The experimental results revealed CBPred could not only explore the original and topological representations of drug disease pairs more deeply, but also exhibited stronger predictive power compared with the baseline models. In addition, in case studies, they confirmed that CBPred can discover potential new disease indications for drugs.

A novel computational model called SAEROF, developed by combining SAE and rotation forest classifiers to predict drug–disease associations, was developed by Jiang et al. (Fig. 8a).[80] Similar to their previous work with SKCNN, they also extracted two forms of drug similarity and disease similarity from the drug–disease association network, but with the difference that the sigmoid kernel similarity of drugs and diseases was replaced by Gaussian interaction profile kernel similarity. Then, they used SAE to learn meaningful feature representations of drugs and diseases by taking the combined similarity descriptors of drugs and diseases as input. Given that the rotation forest method has the advantages of reducing overfitting problem, resisting noise, and insensitivity to abnormal outliers, it was adopted as a classifier to process the extracted features and then gave a predicted score for a drug–disease association. Compared with SKCNN, SAEROF can effectively learn relatively sparse features of drugs and diseases because of the introduction of SAE, showing higher prediction accuracy.

There have been also VAE-based deep-learning frameworks reported for *in silico* drug repositioning, such as DeepDR proposed by Zeng et al.[81] In DeepDR (Fig. 8b), one drug–disease, one drug–side-effect, one drug–target, one drug–drug interaction network, and six drug–drug similarity networks were included. Specially, they fused positive pointwise mutual information (PPMI) matrices representing each network by using the multimodal deep AE (MDA) to obtain high-quality features of drugs. Then, the learned drug features together with known drug–disease associations were fed into a collective VAE (cVAE) to infer new drug-disease associations. Given that DeepDR captures complex topological patterns across different data sources and effectively overcomes the sparsity of drug–disease associations, it achieved state-of-the-art performance and outperformed the baseline methods (i.e., DTINet, KBMF, and SVM). DeepDR was successfully applied for repurposing US Food and Drug Administration (FDA)-approved drugs to treat Alzheimer's disease (e.g., risperidone and aripiprazole) and Parkinson's disease (e.g., methylphenidate and pergolide).

Ge et al. reported a data-driven drug repositioning framework to discover drug candidates for coronavirus 2019 (COVID-19) by combining GCN and statistical analysis techniques.[31] They first constructed a drug–target knowledge graph based on the drug chemical structures, target sequences and their interactions. Then, a network-based knowledge mining algorithm was used to obtain a predicted list of drug candidates by capturing the hidden virus-related feature information from the knowledge network. To improve the prediction accuracy, GCN was used to gather and update the learned feature representation of each node in the graph, which can fully exploit the network topology information. Based on their proposed framework, they discovered that mefuparib (CVL218), a poly-ADP-ribose polymerase 1 (PARP1) inhibitor, could be a potential therapeutic agent for COVID-19 by blocking severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) replication revealed *in vitro* and vivo studies. Moreover, Gysi et al. also used GCN to help identify potential therapeutic agents for COVID-19[83]; the designed GCN model was trained on the heterogeneous graphs, which comprised protein–protein interactions, DTIs, disease–protein associations, and drug–disease associations. Several existing drugs in clinical studies for the treatment of COVID-19 were successfully predicted, such as chloroquine and ritonavir.

By integrating drug–exposure expression profiles and drug–drug links information to find new drugs for breast cancer,[84] Cui et al. established a GNN model called GraphRepur. They collected the drug–exposure gene expression data of genes differentially expressed in breast cancer cells to represent drug signatures, and obtained links information between drugs (e.g., positive and negative drugs for breast cancer) from the STITCH database, including the drug–drug interaction, similarity, and activity. Then, a drug–drug interaction graph was constructed based on the above information, in which drug signatures were used as the node features. GNN took the interaction graph as input and, as output, the possibility of repurposing each drug for breast cancer. By comparison, GraphRepur showed better performance, outperforming previous state-of-the-art approaches and some classic ML methods (e.g., DeepDR, GCN, SVM, and RF). The high-ranked drugs have been validated as new approaches for the treatment of breast cancer, further proving the effectiveness of this method.

GNN methods are mainly used for the prediction of molecular interactions based on the direct similarity between interacting nodes but fail to capture the indirect similarity,[82] resulting in the low utilization rate of molecular interaction networks. Hence, Huang et al. injected skip similarity representing the second-order molecular interaction into the GNN and developed the prediction model SkipGNN (Fig. 8c),[82] which integrates direct similarity and skip similarity between molecules to predict a variety of molecular interactions. They first used the original molecular interaction network to construct the skip graph, which represents the second-order molecular interaction network containing the skip similarity. Then, they constructed two GNNs based on the original graph and the skip graph, respectively to update each node features. Notably, they aggregated the corresponding node features learned from the original graph and the skip graph through the designed iterative fusion scheme to obtain the final feature representation of each node. Subsequently, the features of two target nodes were combined to feed into the FC layer as a binary classifier to output their interaction probability. The authors found that skip similarity allowed SkipGNN to capture more successfully the structure of interaction networks, making more effective use of the molecular interaction network. Evaluation results on four types of interaction network showed that SkipGNN achieved remarkable and robust performance even when the network was noisy and highly incomplete.

In addition, real-world data (RWD) in healthcare, such as EHR, clinical trial data and postmarketing surveillance data, have also been considered promising data sources for drug repositioning. For instance, Liu et al. established an efficient LSTM-based framework to infer drug candidates for a given disease by analysing patients' diagnosis, prescription, and demographic information.[85] They first extracted a list of potential repurposing drug ingredients from patient data for candidates screening. Then, they selected three categories of patients' observational data [demographic characteristics (i.e., age and gender), diagnosis codes, and prescription medication] to be fed into a LSTM-based model with an attention mechanism to evaluate the drug treatment effects on the disease outcome. Drugs with significant beneficial effects are considered as repurposed drug candidates. Comparisons with several existing preclinical drug-repositioning methods showed that the proposed framework performed better at correcting biases and estimating treatment effects, and retaining interpretability for recognizing important confounding. The authors successfully identified several old drugs and drug combinations with respect to coronary artery disease (CAD) by using the proposed framework, which demonstrated the effectiveness of their method in drug repurposing.

Challenges and perspectives

DL frameworks have unique and outstanding advantages over traditional ML methods in handling drug repositioning tasks, as described above, particularly in processing different types of input data for drug repositioning according to the nature of their architecture. For example, CNN shows superior performance in processing image data, such as distance map and grid-style representation; RNN is applied specifically to handle sequence data, such as SMILES, fingerprints of ligand, and sequence of target for DTI prediction; GNN aims to solve the problem of graph-structure data, which is more suitable for identifying new entity associations from complicated heterogeneous networks (e.g., drug–target/gene or drug–disease networks) compared with other DL methods. Nevertheless, there is still room for improvement, and challenges remain.

First, despite enough input data, the performance and accuracy of the DL frameworks largely suffer from unbalanced and poor-quality input data. For example, a network designed to predict the binding affinity is commonly trained with the input labels (e.g. K_d, IC_{50}, or EC_{50} values) from different bioassays. Such experimental values can be inconsistent in terms of different experiment conditions and standards, and even some data might be incorrect because of signal interference or other factors. Moreover, as for the network-based drug repositioning tasks, the established networks (e.g., drug-related or disease-related) mainly rely on large-scale data from publicly available databases and literature, in which the quality of data is not guaranteed and the data set might be unbalanced because of the lack of negative samples of drug–disease pairs. The predicting ability of network-based drug repositioning models is also limited to the number of training data. How to choose high-quality input data and how to make full use of the available drug–target–disease-related resources (e.g., preclinical information, clinical trial information, and postmarketing surveillance data) to enrich their feature descriptors remain challenges. In our opinion, input data from specialized or manually curated database are recommended, and unclear or inconsistent data should be discarded, rather than kept for the breadth of data.

Another challenge is the lack of fully appropriate and sophisticated feature engineering methods, which is another crucial factor influencing model performance. For instance, K_{DEEP} is designed to grasp the spatial relationships of the proteins and ligands using 3D grid representation. However, there are significant costs associated with the huge amount of dummy and useless grids, which slow the training process. DrugVQA provides an efficient way to manage the 3D structure information representation by using distance mapping and achieves a desirable performance. Deeper understanding of protein–ligand interactions and use of multiple data forms will help obtain more reliable engineering data. Furthermore, how to efficiently integrate these multiple types of feature from diverse heterogeneous networks is also a challenge for network-based methods. Some researchers have proposed different feature fusion methods to obtain highly informative feature representation. For instance, the multimodal deep AE was used to integrate each network to obtain high-quality features of drugs; the K-Nearest Neighbors algorithm was also applied to consolidate the given subsets of drug-related and disease-related similarity matrices into two comprehensive matrices for drug and disease. Thus, it is desirable to engineer available data by probing intrinsic nature or associations to achieve model improvement.

Last but not least, although DL models have many advantages, there are still problems during modeling. To enhance model performance and search for the optimal hyperparameters (e.g., learning rate, batch size, layer number, and nonlinear function), hyperparameter optimization is a necessary but especially time-consuming step. Efficient optimization algorithms, such as Bayesian Optimization,[86–87] are able to boost the searching process. In addition, the overfitting problem is still not completely overcome because of the high complexity of model architecture or the limited amount of data. Focusing on these points, researchers need to develop more flexible models to adapt for multiple tasks under different conditions. The lack of model interpretability has always been considered as a limiting factor for their use, such as in in drug-repositioning tasks. Recently, attention mechanisms were proposed to provide explanations of prediction results and enhance the learning ability of models. Different DL models are suitable for different application tasks. For drug repurposing and related tasks, the networks of DL models must be well designed to fit the protein and ligand system. Fundamentally, to better solve drug reposition problems, more efforts, such as on the basic theory of drug–target/disease associations and DL are necessary. We believe that the constant development of new DL algorithms, coupled with advances or achievements in structural biology, medicinal chemistry, biochemistry, and other fields, will improve the efficiency of DL-based target prediction and drug repositioning.

Declaration of interests

The authors declare no conflicts of interest.

Acknowledgments

The authors would like to acknowledge financial support from the National Natural Science Foundation of China (82122065, 82073698, and 81874291), 111 project (B18035), and the Sichuan Science and Technology Program (2018HH0100).

References

1 S. Pushpakom, F. Iorio, P.A. Eyers, K.J. Escott, S. Hopper, A. Wells, et al., Drug repurposing: progress, challenges and recommendations, Nat Rev Drug Discov 18 (2019) 41–58.

2 T.T. Ashburn, K.B. Thor, Drug repositioning: identifying and developing new uses for existing drugs, Nat Rev Drug Discov 3 (2004) 673–683.

3 R. Xiong, L.K. Zhang, S.L. Li, Y. Sun, M.Y. Ding, Y. Wang, et al., Novel and potent inhibitors targeting DHODH are broad-spectrum antivirals against RNA viruses including newly-emerged coronavirus SARS-CoV-2, Prot Cell 11 (2020) 723–739.

4 K. Hu, M.M. Wang, Y. Zhao, Y.T. Zhang, T. Wang, Z.S. Zheng, et al., A small-scale medication of leflunomide as a treatment of COVID-19 in an open-label blank-controlled clinical trial, Virol Sin 35 (2020) 725–733.

5 F. Cantini, L. Niccoli, C. Nannini, D. Matarrese, M.E.D. Natale, D. Lotti, et al., Beneficial impact of baricitinib in COVID-19 moderate pneumonia; multicentre study, J Infect 81 (2020) 647–679.

6 H. Chugh, A. Awasthi, Y. Agarwal, R.K. Gaur, G. Dhawan, R. Chandra, A comprehensive review on potential therapeutics interventions for COVID-19, Eur J Pharmacol 890 (2021) 173741.

7 S.S. Jean, P.R. Hsueh, Old and re-purposed drugs for the treatment of COVID-19, Exp Rev Anti-Infect Ther 18 (2020) 843–847.

8 T.N. Jarada, J.G. Rokne, R. Alhajj, A review of computational drug repositioning: strategies, approaches, opportunities, challenges, and directions, J Cheminformatics 12 (2020) 46–68.

9 J.T. Du, J. Guo, D.W. Kang, Z.H. Li, G. Wang, J.B. Wu, et al., New techniques and strategies in drug discovery, Chin Chem Lett 31 (2020) 1695–1708.

10 Y.Z. Chen, D.G. Zhi, Ligand-protein inverse docking and its potential use in the computer search of protein targets of a small molecule, Prot Struct Funct Bioinf 43 (2001) 217–226.

11 H.L. Li, Z.T. Gao, L. Kang, H.L. Zhang, K. Yang, K.Q. Yu, et al., TarFisDock: a web server for identifying drug targets with docking approach, Nucl Acids Res 34 (2006) W219–W224.

12 S.R. Ellingson, J.C. Smith, J. Baudry, VinaMPI: facilitating multiple receptor high-throughput virtual docking on high-performance computers, J Comput Chem 34 (2013) 2212–2221.

13 J.C. Wang, P.Y. Chu, C.M. Chen, J.H. Lin, idTarget: a web server for identifying protein targets of small chemical molecules with robust scoring functions and a divide-and-conquer docking approach, Nucl Acids Res 40 (2012) W393–W399.

14 K.T. Schomburg, S. Bietz, H. Briem, A.M. Henzler, S. Urbaczek, M. Rarey, Facing the challenges of structure-based target prediction by inverse virtual screening, J Chem Inf Model 54 (2014) 1676–1686.

15 S. Dakshanamurthy, N.T. Issa, S. Assefnia, A. Seshasayee, O.J. Peters, S. Madhavan, et al., Predicting new indications for approved drugs using a proteochemometric method, J Med Chem 55 (2012) 6832–6848.

16 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.

17 M.J. Keiser, V. Setola, J.J. Irwin, C. Laggner, A.I. Abbas, S.J. Hufeisen, et al., Predicting new molecular targets for known drugs, Nature 462 (2009) 175–181.

18 J.Y. Gong, C.Q. Cai, X.F. Liu, X. Ku, H.L. Jiang, D.Q. Gao, et al., ChemMapper: a versatile web server for exploring pharmacology and chemical structure association based on molecular 3D similarity method, Bioinformatics 29 (2013) 1827–1829.

19 V.I. Pérez-Nueno, V. Venkatraman, L. Mavridis, D.W. Ritchie, Detecting drug promiscuity using gaussian ensemble screening, J Chem Inf Model 52 (2012) 1948–1961.

20 Q.Q. Dai, Y.H. Yan, X.L. Ning, G. Li, J.L. Yu, J. Deng, et al., AncPhore: a versatile tool for anchor pharmacophore steered drug discovery with applications in discovery of new inhibitors targeting metallo-β-lactamases and indoleamine/tryptophan 2,3-dioxygenases, Acta Pharm Sin B 11 (2021) 1931–1946.

21 X.F. Liu, S.S. Ouyang, B. Yu, Y.B. Liu, K. Huang, J.Y. Gong, et al., PharmMapper server: a web server for potential drug target identification using pharmacophore mapping approach, Nucl Acids Res 38 (2010) W609–W614.

22 X. Wang, Y.H. Shen, S.W. Wang, S.L. Li, W.L. Zhang, X.F. Liu, et al., PharmMapper 2017 update: a web server for potential drug target identification with a comprehensive target pharmacophore database, Nucl Acids Res 45 (2017) W356–W360.

23 G.B. Li, Z.J. Yu, S. Liu, L.Y. Huang, L.L. Yang, C.T. Lohans, et al., IFPTarget: a customized virtual target identification method based on protein–ligand interaction fingerprinting analyses, J Chem Inf Model 57 (2017) 1640–1651.

24 X.K. Zhou, M.B. Wu, Y.M. Xie, G.B. Li, T. Li, R. Xie, et al., Identification of Glycine Receptor α3 as a colchicine-binding protein, Front Pharmacol 9 (2018) 1238.

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

26 S.Q. Yang, Q. Ye, J.J. Ding, M.Z. Yin, A.P. Lu, X. Chen, et al., Current advances in ligand-based target prediction, WIREs Comput Mol Sci 11 (2021) 1–21.

27 A. Lavecchia, Deep learning in drug discovery: opportunities, challenges and future prospects, Drug Discov Today 24 (2019) 2017–2032.

28 J.M. Stokes, K. Yang, K. Swanson, W.G. Jin, A. Cubillos-Ruiz, N.M. Donghia, et al., A deep learning approach to antibiotic discovery, Cell 180 (2020) 688–702.

29 B.R. Li, C. Dai, L.J. Wang, H.L. Deng, Y.Y. Li, Z. Guan, et al., A novel drug repurposing approach for non–small cell lung cancer using deep learning, PLoS ONE 15 (2020) e0233112.

30 P. Richardson, I. Griffin, C. Tucker, D. Smith, O. Oechsle, A. Phelan, et al., Baricitinib as potential treatment for 2019-nCoV acute respiratory disease, Lancet 395 (2020) e30–e31.

31 Y.Y. Ge, T.Z. Tian, S.L. Huang, F.P. Wan, J.X. Li, S.Y. Li, et al., An integrative drug repositioning framework discovered a potential therapeutic agent targeting COVID-19, Signal Transduct Target Ther 6 (2021) 165–180.

32 G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.

33 G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput 18 (2006) 1527–1554.

34 V. Sze, Y. Chen, T. Yang, J.S. Emer, Efficient Processing of Deep Neural Networks: A Tutorial and Survey, Proc IEEE 105 (2017) 2295–2329.

35 H. Bourlard, Y.J.B.C. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biol Cybern 59 (2004) 291–294.

36 P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, in: Extracting and composing robust features with denoising autoencoders, Association for Computing Machinery, New York, 2008, pp. 1096–1103.

37 R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, Convolutional neural networks: an overview and application in radiology, Insights Imaging 9 (2018) 611–629.

38 R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R.X. Gao, Deep learning and its applications to machine health monitoring, Mech Syst Signal Pr 115 (2019) 213–237.

39 Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48.

40 O. Olurotimi, Recurrent neural network training with feedforward complexity, IEEE Trans Neural Netw Learn Syst 5 (1994) 185–197.

41 S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput 9 (1997) 1735–1780.

42 K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning phrase representations using RNN encoder–decoder for statistical machine translation, ArXiv arXiv (2014) 1406.1078.

43 M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans Signal Process 45 (1997) 2673–2681.

44 Z. Huang, W. Xu, K. Yu, Bidirectional LSTM–CRF models for sequence tagging, ArXiv arXiv (2015) 1508.01991.

45 K. Al-Thelaya, E.-S. El-Alfy, Evaluation of bidirectional LSTM for short-and long-term stock market prediction, in: 9th International Conference on Information and Communication Systems (ICICS), IEEE, New York, 2018, pp. 151–156.

46 M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, in: Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, IEEE, New York, 2005, pp. 729–734.

47 F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans Neural Netw Learn Syst 20 (2009) 61–80.

48 G.B. Li, L.L. Yang, W.J. Wang, L.L. Li, S.Y. Yang, ID-Score: a new empirical scoring function based on a comprehensive set of descriptors related to protein–ligand interactions, J Chem Inf Model 53 (2013) 592–600.

49 M.A. Rezaei, Y. Li, D. Wu, X. Li, C. Li, Deep learning in drug design: protein–ligand binding affinity prediction, IEEE/ACM Trans Comput Biol Bioinform (2020), https://doi.org/10.1109/TCBB.2020.3046945 [Published online December 23, 2020].

50 H. Cho, E. Lee, I. Choi, Layer-wise relevance propagation of InteractionNet explains protein–ligand interactions at the atom level, Sci Rep 10 (2020) 21155–21166.

51 M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, D.R. Koes, Protein–ligand scoring with convolutional neural networks, J Chem Inf Model 57 (2017) 942–957.

52 D.R. Koes, M.P. Baumgartner, C.J. Camacho, Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise, J Chem Inf Model 53 (2013) 1893–1904.

53 J. Jiménez, M. Škalič, G. Martínez-Rosell, G. De Fabritiis, K(DEEP): protein-ligand absolute binding affinity prediction via 3D-convolutional neural networks, J Chem Inf Model 58 (2018) 287–296.

54 J. Jiménez, S. Doerr, G. Martínez-Rosell, A.S. Rose, G. De Fabritiis, DeepSite: protein-binding site predictor using 3D-convolutional neural networks, Bioinformatics 33 (2017) 3036–3042.

55 F.N. Iandola, M.W. Moskewicz, K. Ashraf, S. Han, W. Dally, K.KJA. SqueezeNet, AlexNet-level accuracy with 50x fewer parameters and <1MB model size, ArXiv (2017). arXiv: 1602.07360.

56 A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Proc Adv Neural Inf Process Syst 25 (2012) 1097–1105.

57 F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, New York, 2017, pp. 1800–1807.

58 M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, New York, 2018, pp. 4510–4520.

59 A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., MobileNets: efficient convolutional neural networks for mobile vision applications, ArXiv (2017). arXiv: 1704.04861.

60 N. Ma, X. Zhang, H.-T. Zheng, J. Sun, ShuffleNet V2: practical guidelines for efficient CNN architecture design, Lect Notes Comp Sci 11218 (2018) 116–131.

61 X. Zhang, X. Zhou, M. Lin, J.J. Sun, ShuffleNet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, New York, 2018, pp. 6848–6856.

62 H.C. Jubb, A.P. Higueruelo, B. Ochoa-Montaño, W.R. Pitt, D.B. Ascher, T.L. Blundell, Arpeggio: a web server for calculating and visualising interatomic interactions in protein structures, J Mol Biol 429 (2017) 365–371.

63 E.N. Feinberg, D. Sur, Z.Q. Wu, B.E. Husic, H.H. Mai, Y. Li, S.S. Sun, et al., PotentialNet for molecular property prediction, ACS Cent Sci 4 (2018) 1520–1530.

64 K. Cho, B.V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., Learning phrase representations using RNN encode' decoder for statistical machine translation, ArXiv (2018). arXiv: 1406.1078.

65 J. Lim, S. Ryu, K. Park, Y.J. Choe, J. Ham, W.Y. Kim, Predicting Drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation, J Chem Inf Model 59 (2019) 3981–3988.

66 G. Montavon, A. Binder, S. Lapuschkin, W. Samek, K. Müller, Layer-wise relevance propagation: an overview, Lect Notes Comp Sci 11700 (2019) 193–209.

67 H. Öztürk, A. Özgür, E. Ozkirimli, DeepDTA: deep drug-target binding affinity prediction, Bioinformatics 34 (2018) i821–i829.

68 W. Torng, R.B. Altman, Graph convolutional neural networks for predicting drug-target interactions, J Chem Inf Model 59 (2019) 4131–4149.

69 S. Zheng, Y. Li, S. Chen, J. Xu, Y. Yang, Predicting drug–protein interaction using quasi-visual question answering system, Nat Mach Intell 2 (2020) 134–140.

70 M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, et al., Deep-learning-based drug–target interaction prediction, J Proteome Res 16 (2017) 1401–1409.

71 M. Karimi, D. Wu, Z. Wang, Y. Shen, DeepAffinity: interpretable deep learning of compound protein affinity through unified recurrent and convolutional neural networks, Bioinformatics 35 (2019) 3329–3338.

72 H. Eslami Manoochehri, M. Nourani, Drug-target interaction prediction using semi-bipartite graph model and deep learning, BMC Bioinf 21 (2020) 248–263.

73 J. Peng, J. Li, X. Shang, A learning-based method for drug-target interaction prediction based on feature representation learning and deep neural network, BMC Bioinform 21 (2020) 394–406.

74 W. Wang, X. Yang, C. Wu, C. Yang, CGINet: graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph, BMC Bioinform 21 (2020) 544–550.

75 N. Zong, H. Kim, V. Ngo, O. Harismendy, Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations, Bioinformatics 33 (2017) 2337–2344.

76 H. Liu, W. Zhang, Y. Song, L. Deng, S. Zhou, HNet-DNN: inferring new drug–disease associations with deep neural network based on heterogeneous network features, J Chem Inf Model 60 (2020) 2367–2376.

77 T. Jarada, J. Rokne, R. Alhajj, SNF-NN: computational method to predict drug-disease interactions using similarity network fusion and neural networks, BMC Bioinf 22 (2021) 28–47.

78 P. Xuan, Y. Ye, T. Zhang, L. Zhao, C. Sun, Convolutional neural network and bidirectional long short-term memory-based method for predicting drug-disease associations, Cells 8 (2019) 705–719.

79 H.J. Jiang, Z.H. You, Y.A. Huang, Predicting drug-disease associations via sigmoid kernel-based convolutional neural networks, J Transl Med 17 (2019) 382–392.

80 H.J. Jiang, Y.A. Huang, Z.H. You, SAEROF: an ensemble approach for large-scale drug-disease association prediction by incorporating rotation forest and sparse autoencoder deep neural network, Sci Rep 10 (2020) 4972–4982.

81 X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, F. Cheng, deepDR: a network-based deep learning approach to in silico drug repositioning, Bioinformatics 35 (2019) 5191–5198.

82 K. Huang, C. Xiao, L.M. Glass, M. Zitnik, J. Sun, SkipGNN: predicting molecular interactions with skip-graph networks, Sci Rep 10 (2020) 21092–21107.

83 D.M. Gysi, Í. Do Valle, M. Zitnik, A. Ameli, X. Gan, O. Varol, et al., Network medicine framework for identifying drug repurposing opportunities for COVID-19, ArXiv (2020). arXiv: 2004.07229.

84 C. Cui, X. Ding, D. Wang, L. Chen, F. Xiao, Y. Xu, et al., Drug repurposing against breast cancer by integrating drug-exposure expression profiles and drug–drug links based on graph neural network, Bioinformatics 37 (2021) 2930–2937.

85 R. Liu, L. Wei, P. Zhang, A deep learning framework for drug repurposing via emulating clinical trials on real-world patient data, Nat Mach Intell 3 (2021) 68–75.

86 J. Snoek, H. Larochelle, R. Adams, Practical Bayesian optimization of machine learning algorithms, Adv Neural Inf Process Syst 25 (2012) 2951–2959.

87 P. Murugan, Hyperparameters optimization in deep convolutional neural network/Bayesian approach with Gaussian process prior, ArXiv (2017). arXiv: 1712.07233.