Computers in Biology and Medicine 149 (2022) 105984

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

DeepPSE: Prediction of polypharmacy side effects by fusing deep representation of drug pairs and attention mechanism

Shenggeng Lin [a], Guangwei Zhang [b], Dong-Qing Wei [a,c,d,*], Yi Xiong [a,e,**]

[a] State Key Laboratory of Microbial Metabolism, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, 200240, China
[b] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong, 510275, China
[c] Zhongjing Research and Industrialization Institute of Chinese Medicine, Zhongguancun Scientific Park, Meixi, Nayang, Henan, 473006, China
[d] Peng Cheng National Laboratory, Vanke Cloud City Phase I Building 8, Xili Street, Nanshan District, Shenzhen, Guangdong, 518055, China
[e] Shanghai Artificial Intelligence Laboratory, Shanghai, China

ARTICLE INFO

Keywords:
Polypharmacy side effect prediction
Drug-drug interactions
Feature fusion
Self-attention mechanism

ABSTRACT

Polypharmacy (multiple use of drugs) is an effective strategy for combating complex or co-existing diseases. However, a major consequence of polypharmacy is a higher risk of adverse side effects due to drug-drug interactions, which are rare and observed in relatively small clinical testing. Thus, identification of polypharmacy side effects remains challenging. Here, we propose a deep learning-based method, DeepPSE, to predict polypharmacy side effects in an end-to-end way. DeepPSE is composed of two main modules. First, multiple types of neural networks are constructed and fused to learn the deep representation of a drug pair. Second, the encoder block of transformer that includes self-attention mechanism is built to get latent features, which are further fed into the fully connected layer to predict polypharmacy side effects of drug pairs. Further, DeepPSE is compared with five baseline or state-of-the-art methods on a benchmark dataset of 964 types of polypharmacy side effects across 63473 drug pairs. Experimental results demonstrate that DeepPSE achieves better performance than that of all five methods. The source codes and data are available at https://github.com/ShenggengLin/DeepPSE

1. Introduction

Polypharmacy (i.e., multiple drugs are jointly used) is an effective strategy for combating complex or co-existing diseases [1–4]. A major consequence of polypharmacy to a patient is a much higher risk of side effects due to adverse drug-drug interactions [5–8]. Reliable identification of polypharmacy side effects is challenging because they are rare and observed in relatively small clinical testing. It is practically impossible to experimentally identify the polypharmacy side effects of all possible pairs of drugs. Therefore, it is desirable and urgent to develop computational methods to predict polypharmacy side effects, which is vital to drug discovery and development [9–15].

In recent years, side effect data of single drugs or drug combinations are collected from relevant literature, clinical trials, laboratory studies and electronic medical records to construct databases, which facilitate the development of computational methods for predicting polypharmacy side effects. Since 2018, there are several studies to develop data-driven or/and knowledge-driven approaches to predict polypharmacy side effects by deep neural network (DNN) [16], graph convolutional network (GCN) [17] and knowledge graph (KG) representation learning methods [18–21]. The pioneering study by Zitnik et al. [17] constructed the benchmark dataset of 964 commonly occurring types of polypharmacy side effects across 63473 drug combinations. Then, they formulated polypharmacy side effect modeling as a multi-relational link prediction problem on a multimodal graph consisting of drug, protein and side effect relationships. They proposed Decagon to predict what will the exact type of the side effect be for a given pair of drugs by using GCN in an end-to-end way, based on a multimodal graph of protein-protein interactions, drug-target interactions and DDIs, where each side effect is an edge of a different type. This architecture becomes a baseline for several other state-of-the-art methods for polypharmacy side effect prediction.

* Corresponding author. State Key Laboratory of Microbial Metabolism, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, 200240, China.
** Corresponding author. State Key Laboratory of Microbial Metabolism, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, 200240, China.
E-mail addresses: dqwei@sjtu.edu.cn (D.-Q. Wei), xiongyi@sjtu.edu.cn (Y. Xiong).

https://doi.org/10.1016/j.compbiomed.2022.105984
Received 22 June 2022; Received in revised form 17 July 2022; Accepted 14 August 2022
Available online 18 August 2022
0010-4825/© 2022 Elsevier Ltd. All rights reserved.

Based on the baseline model Decagon, Wang et al. [22] added drug-enzyme interactions to the multimodal graph to improve prediction of polypharmacy side effects. Instead of viewing the graph as a whole, Xu et al. [23] proposed a tri-graph information propagation model to view the multi-modal biomedical graph as three subgraphs. This method embeds proteins and drugs into different spaces of possibly different dimensions, rather than the same space and dimensions. Later, several studies are developed for interpretable prediction of polypharmacy side effects by using graph feature attention network [18,20]. For example, Yao et al. [20] proposed a novel model by further incorporating complex relations of side effects into knowledge graph embeddings. This model can translate and transmit multidirectional semantics with fewer parameters, leading to better scalability in large-scale knowledge graphs. Novacek et al. [24] proposed a new knowledge graph embedding technique that uses multi-part embedding vectors to predict polypharmacy side-effects. Masumshah et al. [16] proposed a neural network-based method for polypharmacy side effect prediction by using feature vectors based on mono side effects, and drug-protein interaction (DPI) information.

Although those methods mentioned above have achieved satisfactory performance, they still have some limitations. For example, the existing studies just concatenate two drug vectors together without trying other ways to fuse the information to represent drug pairs, which may miss the rich information contained in the drug pairs. In order to explore whether different drug fusion methods are beneficial to predict polypharmacy side effects, we propose a novel method, DeepPSE, which predicts polypharmacy side effects based on deep representation of drug pairs and self-attention mechanism. DeepPSE contains two main modules. First, multiple types of neural networks are constructed to learn the deep representation of a drug pair in four different way. Second, the encoder block of transformer [25] that includes self-attention mechanism is built to perform latent feature fusion with the four latent vectors of drug pairs mentioned above and the fifth hidden vector obtained by element-wise addition of these four latent vectors, which are combined and further fed into the fully connected layer to predict polypharmacy side effects.

Furthermore, DeepPSE is compared to five baseline or state-of-the-art methods on a benchmark dataset of 964 types of polypharmacy side effects across 63473 drug pairs. Experimental results demonstrate that our proposed method DeepPSE achieves better performance than that of all five methods. Moreover, our model also proves the effectiveness of the feature fusion of deep representation of drug pairs by various neural networks.

2. Materials and methods

2.1. Datasets

In this work, we used the same benchmark dataset as that constructed by Zitnik et al. [17]. The dataset contains 645 drugs and 964 polypharmacy side effect types. Each drug has two types of features about mono side effects and DPI information. In this section, the polypharmacy side effects, the mono side effects, and the DPIs are presented in details as below.

2.1.1. Polypharmacy side effects

Polypharmacy side effects are collected from the TWOSIDES database [26], which is sourced from the Food and Drug Administration Adverse Event Reporting System (FAERS) and provides a reliable and comprehensive DDIs database with 1317 side effects for 645 drugs across 63473 drug pairs. As in the previous studies on predicting polypharmacy side effects [16,17], we consider 964 polypharmacy side effects which occurred in at least 500 DDIs.

2.1.2. Mono side effects information

The side effects of individual drugs (mono side effects) are obtained from the Side Effects Resource (SIDER) and OFFSIDES databases [27]. The information in the SIDER database is extracted from drug labels, which contains 1556 drugs and 5868 side effects compiled from public documents. The entries in OFFSIDES database are observed during clinical trials, which contain side effects for 1332 drugs and 10097 labels. Like TWOSIDES, OFFSIDES was generated from FAERS that was collected from clinical reports, patients, and drug companies. Finally, 10184 mono side effects of 645 drugs in the TWOSIDES database were obtained by merging and eliminating synonym side effects in the SIDER and OFFSIDES databases.

2.1.3. Drug-protein interactions

DPIs are obtained from the Search Tool for Interactions of Chemicals (STITCH) database [28], which provides the relationship between drugs and target proteins. Using the STITCH database, we obtained interactions between 7795 proteins and 645 drugs in the TWOSIDES database.

2.2. Drug feature vectors

The feature vector for each drug contains a 10184-dimensional mono side effect vector and a 7795-dimensional DPI vector. Due to the large length and sparseness of the feature, the feature extraction is employed to effectively reduce the size of the feature without losing important information [29–31]. As in the previous study [16], the Principle Components Analysis (PCA) is applied on mono side effects and DPIs matrices. The minimum number of the principle components is chosen such that 95% on variance in each matrix is retained. After dimensionality reduction, the drug feature vector consists of a 503-dimensional mono side effect vector and a 22-dimensional DPI vector, as shown in Fig. 1(A).

2.3. Deep representation of drug pairs

Multiple deep neural networks, which include convolutional neural network (CNN), two auto-encoders with self-attention mechanism and a Siamese network [32], are used for fusing and representing drug pairs, as shown in Fig. 1(B). Compared with simply concatenating or fusing the information of a drug pair using only one method, the multiple deep neural networks can capture diverse information from multi-views for deep learning-based models to accurately predict polypharmacy side effects. Next, we will describe them in detail.

CNN has achieved satisfactory performance in computer vision because its convolution operation focuses on local information. In our model, each drug is represented by a *1*525*-dimensional vector. We combine two drug vectors into a *2*525*-dimensional matrix and input it into CNN, whose kernel size is *2*p*. Therefore, the CNN will output a row vector as the latent vector of the drug pair. The row vector is finally fed into the 1-dimensional CNN to obtain the final latent vector (LF1) of the drug pair, as shown in Fig. 1(B.I).

Auto-encoder is an unsupervised neural network model, which is composed of an encoder and a decoder. It can learn the representation hidden in the input data without annotations. The self-attention mechanism is a variant of the attention mechanism, which can focus on important features by assigning different weights to different features. Therefore, we append a self-attention layer before the output layer of the encoder in two different auto-encoders, resulting in two different auto-encoders with self-attention mechanism. A *1*1050*-dimensional vector generated by concatenating two *1*525*-dimensional vectors is input into the first auto-encoder with a self-attention mechanism (named AE1) to perform drug fusion, as shown in Fig. 1(B.II). While the second auto-encoder with a self-attention mechanism (named AE2) is fed with the *1*525*-dimensional vector obtained by element-wise adding two drug vectors, as shown in Fig. 1(B.III). The latent vectors of two auto-encoders (LF2,LF3) are used as the latent vectors of the drug pair.

The Siamese network can reduce the number of parameters in the

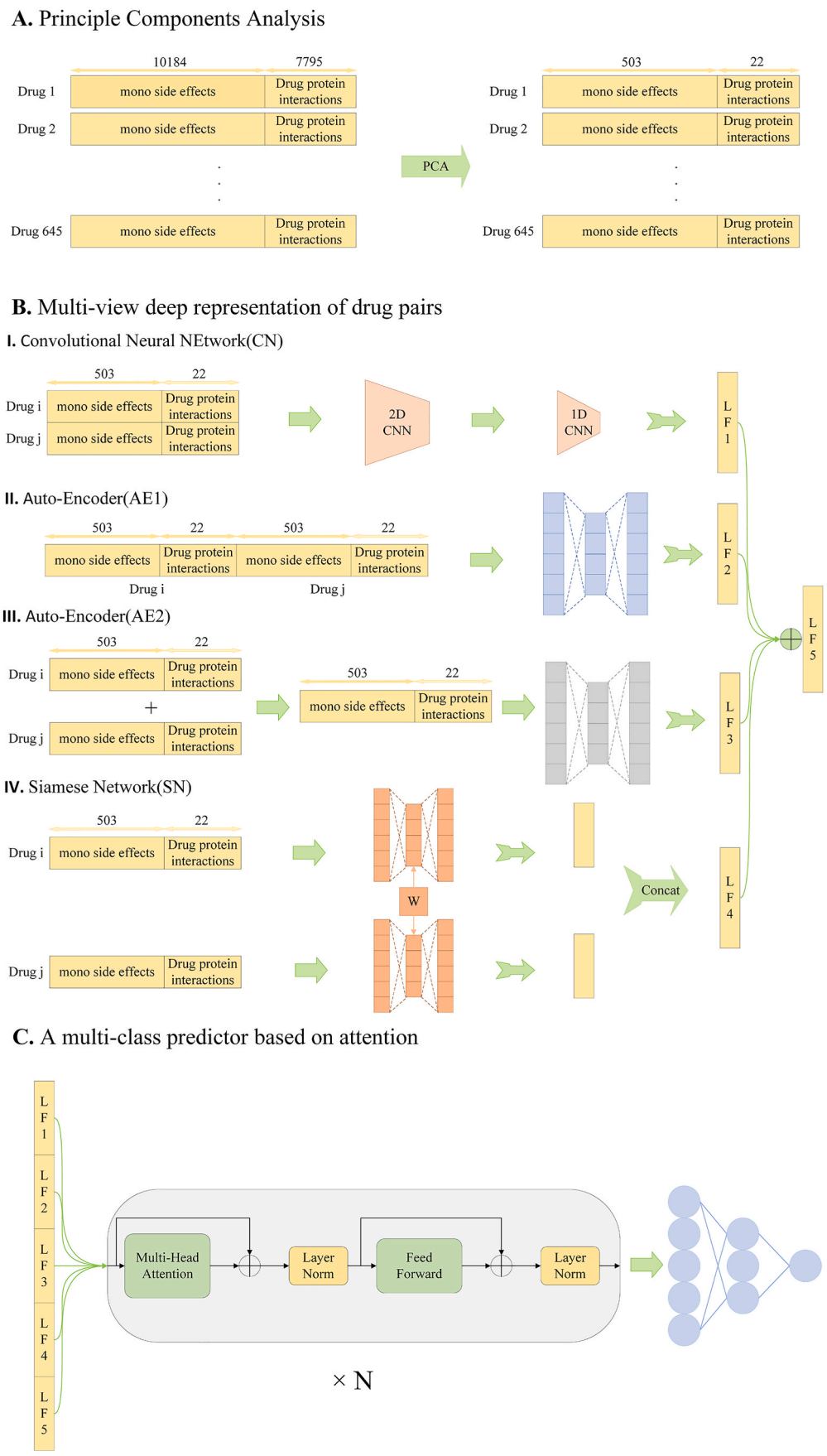

Fig. 1. The workflow of the proposed DeepPSE method.

neural network by sharing weights. In our model, another two auto-encoders are used as sub-networks of the Siamese network (named SN), as shown in Fig. 1(B.IV). We feed two drug vectors into these two auto-encoders of SN to obtain two feature vectors, respectively. These two vectors are finally concatenated as the latent vector (LF4) of the drug pair. The parameters of two auto-encoders are shared so that the latent vector contains the information of drug pairs. Besides, Siamese auto-encoder network takes a single drug as input rather than a drug pair to capture drug-level features, instead of drug-pair-level features.

2.4. Latent feature fusion

We use the encoder structure of the transformer to perform latent feature fusion. The four latent vectors (LF1,LF2,LF3,LF4) are provided by four different methods and network structures mentioned above. Then, we obtain the fifth feature vector (LF5) by element-wise adding these four feature vectors. LF5 can update the parameters of different drug fusion networks at the same time, which increases the communication between different drug fusion networks during the training process. Finally, we concatenate these five vectors as the new feature of a drug pair and feed this new feature into the encoder structure of the transformer, as shown in Fig. 1(C). The encoder structure of the transformer is mainly composed of self-attention layer, layer Normalization, Residual Connections and feed-forward layer.

Different feature vectors have different contributions to prediction of polypharmacy side effects. We feed these five feature vectors represented by the concatenated vector into the multi-head attention module, which is also called the Transformer block, to learn the weight of each vector. The self-attention mechanism will recognize which feature vectors are more important for prediction and give them large weights [33,34]. Residual connection [35] can partially solve the gradient disappearance problem. The gradient can be directly propagated through residual connection during back propagation, which makes the speed of back propagation and gradient update much faster. Therefore, the network structure can be designed very deep. Layer normalization was usually used in two scenarios: after the self-attention layer and after the feed-forward network layer. Its purpose is to ameliorate the "covariate-shift" problem by re-standardizing the calculated vector representations. It can also accelerate the convergence of neural network parameters.

2.5. Loss function

We choose binary cross-entropy loss (BCELoss) as the loss function for each type of polypharmacy side effect, and take their average of all 964 types as the total classification loss.

$$\mathrm{MLoss} = \frac{\sum_{i=1}^{964} \mathrm{BCELoss}_{\text{side effect i}}}{964} \tag{1}$$

In our model, AE1, AE2 and SN are all auto-encoders, so we choose the Mean Squared Error (MSE) loss function for them as the auxiliary loss of classification loss. The binary cross-entropy loss is multiplied by a corresponding classification loss weight (clw) to make the model pay more attention to classification loss. Thus, the total loss function of the model is as follows:

$$\mathrm{loss} = \mathrm{clw} \times \mathrm{MLoss} + \mathrm{MSE}_{\mathrm{AE1}} + \mathrm{MSE}_{\mathrm{AE2}} + \mathrm{MSE}_{\mathrm{SN}} \tag{2}$$

3. Results and discussion

3.1. Experimental settings

The drug pairs associated with each type of side effects are split into training, validation, and test sets. We use 90% of drug pairs for the training set, 5% for the validation set, and 5% for the test set. In order to avoid the randomness of the results, we repeatedly run the procedure 5 times and take the average as the final result.

It is important to emphasize that we did not perform cross-validation on the training set. Instead, we randomly divided the whole dataset into training set, validation set and test set. Since the randomness of the data partition could affect the model evaluation results, we repeated the data set partition five times. Every time we get the training set, validation set and test set, we retrain the model on the training set, tune the parameters through the validation set, and test the model on the test set. We take the average of the five test results as the final performance.

We adopt accuracy (ACC), area under the precision-recall-curve (AUPR), area under the ROC curve (AUC), F1 score, Matthews Correlation Coeffcient (MCC), Precision and Recall as evaluation metrics for model evaluation, which are widely used in machine learning applications [36–45].

3.2. Hyper-parameter setting

The choice of hyper-parameters influences the performance of model. Therefore, we discussed six hyper-parameters: hidden layer dimension of auto-encoders (HLD), self-attention module layers (SML), dropout rating (DR), learning rate (LRA), batch size (BS) and training epochs (TE). These hyper-parameters may have a huge impact on model performance. The hidden layer dimension of auto-encoders and self-attention module layers determine the size and fit ability of the model. A suitable dropout rating can prevent the model from overfitting. The learning rate and batch size determine whether and how quickly the model converges. Training epochs can set a suitable training time for the model. Therefore, we choose these six hyper-parameters for tuning.

We tune the six hyper-parameters in the order of HLD, SML, DR, LRA, BS, TE. While tuning one of the hyper-parameters, the other five hyper-parameters remain unchanged. We did not use grid search to find the optimal hyper-parameter combination. Because there are 4^6 parameter combinations for 6 parameters, this search space is too large for grid search.

We use gaussian error linear unit (GELU) activation function [46] and Adam optimizer [47]. The dropout layer and batch normalization [48] layers are used between the fully connected layers. The metric scores under different configurations are shown in Fig. 2.

As shown in Fig. 2, the performance of our model does not change greatly with the change of hyper-parameters. Almost all metric scores vary within the range of 0.01, which also demonstrates that our model is robust and stable. In the end, we chose 400 for HLD, 4 for SML, 0.5 for DR, 2e-5 for LRA, 2048 for BS and 50 for TE.

3.3. Feature evaluation

In this section, we evaluate the effects of mono side effects and DPIs on prediction performance, respectively. The results of all prediction models are shown in Fig. 3.

As shown in Fig. 3, the prediction performance of the model using only mono side effects are much better than that of the model using only DPIs. This may be due to the dimensionality of DPI features is too small after PCA dimensionality reduction, resulting in poor fitting ability of the model. Moreover, it also shows that mono side effect is an important kind of feature for polypharmacy side effect prediction. When using mono side effects and DPIs for prediction, the prediction results are slightly better than the results of using only mono side effects, but the improvement is not too obvious. Therefore, using more kinds of features may increase the computational cost of the model without significant performance improvement.

Proteins can affect drug transport, absorption, pharmacological effects, toxic side effects and antibiotic drug effects. The targets of many drugs are proteins, and drugs treat diseases by interacting with proteins. Therefore, DPIs are important drug features for prediction of polypharmacy side effects. In our model, the prediction performance of the

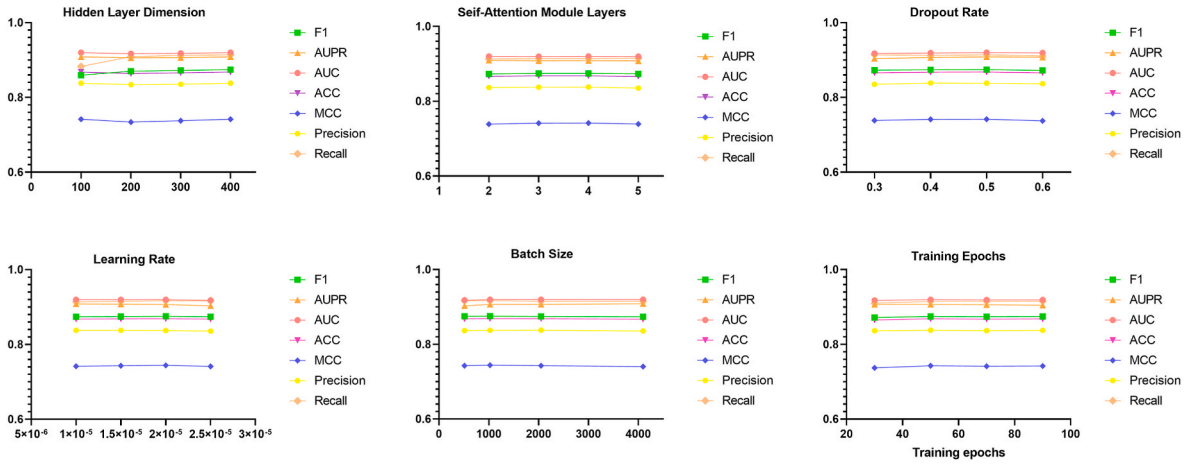

Fig. 2. The metric scores under different hyper-parameters.

model using only DPIs does not work well probably because the dimensionality of DPI features is too small after PCA dimensionality reduction. But that does not mean that DPIs are not important.

Similar mono side effects may correspond to similar targets. Therefore, when two drugs have similar mono side effects, the two drugs are more likely to have the same targets, and polypharmacy side effects are more likely to occur. Therefore, mono side effects are important features for predicting polypharmacy side effects. Our experiments also demonstrate that the model can achieve satisfactory results using only mono side effects. This also demonstrates the importance of the feature of mono side effects.

3.4. Mean squared error loss evaluation

The loss functions of our model include binary cross-entropy loss and mean squared error loss. Since our model is a binary classification model, binary cross-entropy loss is required during training. To verify whether the mean squared error loss is beneficial to the training of the model, we compare the performance of the model with and without the mean squared error loss. The experimental results are shown in Table 1.

According to Table 1, whether or not to use MSE has little effect on the performance of the model. There is little difference in the performance of the model with and without MSE. Therefore, MSE does not play a significant role in the training of our model. That is, our model can be trained well using only binary cross-entropy loss.

3.5. The effect of multiple DNNs for deep representation of drug pairs

In this section, we evaluate the impact of different DNNs of fusing the drug pairs on prediction of polypharmacy side effects. To compare the difference between fusing information in a single way and fusing in multiple ways, several models are built and the metric scores of the models are used to evaluate their predictive power. The results of all prediction models are shown in Table 2.

Among all drug fusions, the element-wise summation of feature vectors of two drugs (AE2) seems to be the most informative and achieves the best performance on all evaluation metrics. It produces an AUC of 0.920 and an AUPR of 0.911. The model which concatenates two drugs into a *1*1050*-dimensional vector (AE1) produces an AUC of 0.903, and the model with feature vectors of two drugs into the SN produces an AUC of 0.907. The model that combines two drug features into a 2*525-dimensional feature vector (CN) gets an AUC of 0.501. The AUC in this drug combination method is low, and this is probably because 2*525-dimensional feature vectors are only suitable for 2D CNN to extract features. The combination of several different drug fusions provides the slight improvement compared with only one single version of drug fusions. The combination of AE2 and CN produces the best AUC (0.930) and achieves the best performance on all evaluation metrics among all combinations of two versions of drug fusion. The combination of AE2, CN and SN achieves the best performance on all evaluation metrics among all combinations of three versions of drug fusion. To sum up, the combination of AE2 and CN performs the best on all evaluation metrics in all combinations of drug fusions.

Overall, the performance of AE2 and SN drug fusion network is better than that of AE1 and CN. The AE2 and SN have in common that they both use an auto-encoder network structure and eliminate the effect of the order of the drugs in the drug pair. It suggests that auto-encoder maybe a good choice for extracting drug features. CNNs performed the worst of all networks, probably because the features of drug pairs and images have different properties. Although we can combine the features of drug pairs into the form of images and use CNNs to extract the features of drug pairs, due to the difference in the properties of drug pairs and images, CNNs cannot achieve good performance.

CN extracts drug pair features through convolution kernels, and it pays more attention to the local information of drug pair features. AE2 extracts drug pair features by directly reducing the dimensionality of the drug pair features, so some information may be lost in the process of dimensionality reduction. Therefore, the drug pair features extracted by CN can be used as an effective supplement to the drug pair features extracted by AE2. The two drug fusion networks extract the features of drug pairs from different views, providing more information for polypharmacy side effect prediction, so the CN + AE2 can achieve better performance.

As mentioned in section 2.3, compared with fusing the information of one drug pair using only one method, multiple views of drug fusion can provide deep learning models with diverse information from different perspectives, which can accurately predict polypharmacy side effects. However, compared with using more number of drug fusion methods, the performance of the model using two drug fusion methods will be better. Because using too many drug fusion methods may make the model too complex and prone to be over-fitting. Therefore, in practical application, we need to comprehensively consider the complexity of the problem and the fitting ability of the model to select an appropriate model to solve the problem.

3.6. Comparison with other methods

In this section, the performance of DeepPSE is benchmarked against 5 well-known methods, which are Decagon, Concatenated drug features, Deep Walk, DEDICOM, and RESCAL, for prediction of 964 polypharmacy side effect types. The AUC and AUPR values of all methods for 964 polypharmacy side effects are shown in Table 3. Because only the

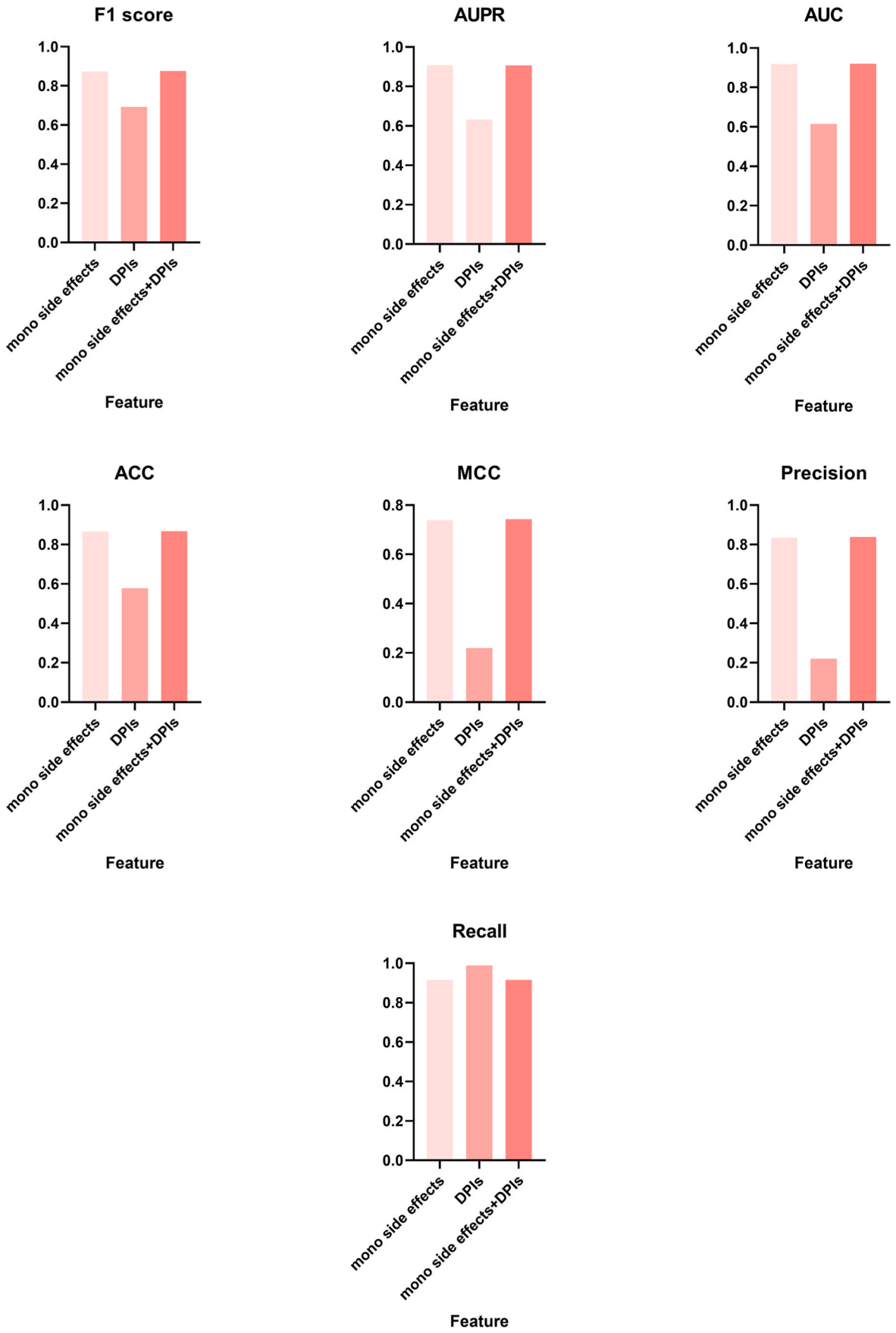

Fig. 3. The effects of mono side effects and DPIs on prediction performance.

Table 1
The performance of MDF-PSE with/without MSE.

	F1 score	AUPR	AUC	ACC	MCC	Precision	Recall
With MSE	0.8855	0.9200	0.9302	0.8804	0.7657	0.8530	0.9250
Without MSE	0.8856	0.9198	0.9300	0.8805	0.7660	0.8532	0.9253

Table 2
The performance of MDF-PSE with different drug fusions.

	F1 score	AUPR	AUC	ACC	MCC	Precision	Recall
AE1	0.856	0.893	0.903	0.848	0.702	0.816	0.900
AE2	0.874	0.911	0.920	0.867	0.741	0.836	0.916
SN	0.862	0.897	0.907	0.855	0.716	0.825	0.902
CN	0.673	0.521	0.501	0.526	0.097	0.515	0.971
AE1+AE2	0.870	0.906	0.917	0.863	0.733	0.833	0.910
AE1+SN	0.868	0.903	0.915	0.862	0.730	0.833	0.906
AE1+CN	0.859	0.896	0.907	0.851	0.709	0.819	0.903
AE2+SN	0.883	0.917	0.928	0.878	0.761	0.848	0.921
AE2+CN	**0.885**	**0.920**	**0.930**	**0.880**	**0.766**	**0.853**	**0.925**
SN + CN	0.872	0.908	0.919	0.866	0.738	0.837	0.910
AE1+AE2+SN	0.875	0.908	0.920	0.868	0.742	0.837	0.917
AE1+AE2+CN	0.871	0.906	0.917	0.865	0.736	0.834	0.911
AE1+SN + CN	0.869	0.903	0.915	0.862	0.730	0.832	0.909
AE2+SN + CN	**0.885**	0.918	0.929	0.879	0.764	0.849	0.924
AE1+AE2+SN + CN	0.875	0.907	0.920	0.868	0.743	0.838	0.915

Table 3
The average of AUC, AUPR for 964 polypharmacy side efects prediction.

Method	AUC	AUPR
MDF-PSE	0.930	0.920
Decagon	0.874	0.825
Concatenated drug features	0.793	0.764
DeepWalk	0.761	0.737
DEDICOM	0.705	0.637
RESCAL	0.693	0.613

source code and implementation of Decagon are available, we repeat Decagon 5 times and the obtained results are very similar to the reported results of the Decagon method. In Table 3, we mention the average of the obtained results for the Decagon method and the reported performance of other methods whose source code we do not have by using Table 2 in previous work [17]. In previous studies [16,17], only AUC and AUPR for these methods are reported, while other metrics are not, so we can only compare the AUC and AUPR of DeepPSE with those of these methods. According to Table 3, DeepPSE algorithm is 5.6% and 9.5% better than the Decagon algorithm in terms of AUROC and AUPRC, respectively.

For more evaluation, we compare DeepPSE with Decagon in detail, and the specific results are shown in Table 4. According to Table 4, DeepPSE outperforms about 3.5%, 4.9%, and 8.1% against Decagon based on F-score, ACC, and MCC criteria, respectively.

4. Conclusion

We proposed a polypharmacy side effect prediction model by fusing multi-view of deep representation of drug pairs and the attention mechanism, and proved the effectiveness and robustness of our model and effects of different features for polypharmacy side effect prediction. In addition, we also proved that the performance of the model using two drug fusion methods rather than using more drug fusion methods will be better. Because using too many drug fusion methods may make the model too complex and prone to be over-fitting. Experimental results have proved that our proposed model is better than the state-of-the-art models. Therefore, we provide a promising approach for polypharmacy side effect prediction.

Table 4
The results for MDF-PSE and Decagon methods for 964 side effects.

Method	F1 score	AUPR	AUC	ACC	MCC	Precision	Recall
MDF-PSE	0.885	0.920	0.930	0.880	0.766	0.853	0.925
Decagon	0.850	0.825	0.874	0.831	0.685	0.771	0.950

Statement

Author contributions: Conceptualization and design, data acquisition and analysis, methodology, Shenggeng Lin; investigation, writing-original and draft preparation, Guangwei Zhang; writing-review, editing and project administration Yi Xiong; funding acquisition, Yi Xiong and Dong-Qing Wei.

Data availability

The source codes and data are available at https://github.com/ShenggengLin/DeepPSE

Funding

This work is supported by grants from the National Science Foundation of China (Grant Nos. 62172274, 32070662, 61832019, 32030063), the Science and Technology Commission of Shanghai Municipality (Grant No. 19430750600), and SJTU JiRLMDS Joint Research Fund and Joint Research Funds for Medical and Engineering and Scientific Research at Shanghai Jiao Tong University (YG2021ZD02). The computations were partially performed at the Pengcheng Lab and the Center for High-Performance Computing, Shanghai Jiao Tong University.

Declaration of competing interest

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

References

[1] X. Sun, Y. Zhang, Y. Zhou, et al., NPCDR: natural product-based drug combination and its disease-specific molecular regulation, Nucleic Acids Res. 50 (2022) D1324–D1333.

[2] H. Liu, W. Zhang, B. Zou, et al., DrugCombDB: a comprehensive database of drug combinations toward the discovery of combinatorial therapy, Nucleic Acids Res. 48 (2020) D871–D881.

[3] M.P. Menden, D. Wang, M.J. Mason, et al., Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen, Nat. Commun. 10 (2019) 2674.

[4] M. Tyers, G.D. Wright, Drug combinations: a strategy to extend the life of antibiotics in the 21st century, Nat. Rev. Microbiol. 17 (2019) 141–155.

[5] Y.H. Feng, S.W. Zhang, Q.Q. Zhang, et al., deepMDDI: a deep graph convolutional network framework for multi-label prediction of drug-drug interactions, Anal. Biochem. 646 (2022), 114631.

[6] Y. Deng, X. Xu, Y. Qiu, et al., A multimodal deep learning framework for predicting drug-drug interaction events, Bioinformatics 36 (2020) 4316–4322.

[7] S. Lin, Y. Wang, L. Zhang, et al., MDF-SA-DDI: predicting drug-drug interaction events based on multi-source drug fusion, multi-source feature fusion and transformer self-attention mechanism, Briefings Bioinf. 23 (2022).

[8] J. Yin, F. Li, Y. Zhou, et al., INTEDE: interactome of drug-metabolizing enzymes, Nucleic Acids Res. 49 (2021) D1233–D1243.

[9] X.Q. Ru, X.C. Ye, T. Sakurai, et al., NerLTR-DTA: drug-target binding affinity prediction based on neighbor relationship and learning to rank, Bioinformatics 38 (2022) 1964–1971.

[10] L. Shen, F. Liu, L. Huang, et al., VDA-RWLRLS: an anti-SARS-CoV-2 drug prioritizing framework combining an unbalanced bi-random walk and Laplacian regularized least squares, Comput. Biol. Med. 140 (2021), 105119.

[11] Y. Chu, X. Shan, T. Chen, et al., DTI-MLCD: predicting drug-target interactions using multi-label learning with community detection method, Briefings Bioinf. 22 (2021).

[12] X. Zeng, X. Tu, Y. Liu, et al., Toward better drug discovery with knowledge graph, Curr. Opin. Struct. Biol. 72 (2021) 114–126.

[13] Y. Ding, J. Tang, F. Guo, et al., Identification of drug-target interactions via multiple kernel-based triple collaborative matrix factorization, Briefings Bioinf. 23 (2022).

[14] Y. Chu, A.C. Kaushik, X. Wang, et al., DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features, Briefings Bioinf. 22 (2021) 451–462.

[15] Y. Deng, Y. Qiu, X. Xu, et al., META-DDIE: predicting drug-drug interaction events with few-shot learning, Briefings Bioinf. 23 (2022).

[16] R. Masumshah, R. Aghdam, C. Eslahchi, A neural network-based method for polypharmacy side effects prediction, BMC Bioinf. 22 (2021) 385.

[17] M. Zitnik, M. Agrawal, J. Leskovec, Modeling polypharmacy side effects with graph convolutional networks, Bioinformatics 34 (2018) i457–i466.

[18] S. Bang, J.H. Jhee, H. Shin, Polypharmacy side-effect prediction with enhanced interpretability based on graph feature attention network, Bioinformatics 37 (2021) 2955–2962.

[19] Y.J. Chen, T.F. Ma, X.X. Yang, et al., MUFFIN: multi-scale feature fusion for drug-drug interaction prediction, Bioinformatics 37 (2021) 2651–2658.

[20] J.F. Yao, W. Sun, Z.Q. Jian, et al., Effective knowledge graph embeddings based on multidirectional semantics relations for polypharmacy side effects prediction, Bioinformatics 38 (2022) 2315–2322.

[21] Y. Yu, K.X. Huang, C. Zhang, et al., SumGNN: multi-typed drug interaction prediction via efficient knowledge graph summarization, Bioinformatics 37 (2021) 2988–2995.

[22] R. Wang, T. Li, Z. Yang, et al., Predicting Polypharmacy Side Effects Based on an Enhanced Domain Knowledge Graph, Springer International Publishing, Cham, 2020, pp. 89–103.

[23] H. Xu, S. Sang, H. Lu, Tri-graph Information Propagation for Polypharmacy Side Effect Prediction, 2020, 10516 arXiv:2001.

[24] V. Novacek, S.K. Mohamed, Predicting polypharmacy side-effects using knowledge graph embeddings, AMIA Jt Summits Transl Sci Proc 2020 (2020) 449–458.

[25] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 30 (Nips 2017).

[26] N.P. Tatonetti, P.P. Ye, R. Daneshjou, et al., Data-driven prediction of drug effects and interactions, Sci. Transl. Med. 4 (2012).

[27] M. Kuhn, I. Letunic, L.J. Jensen, et al., The SIDER database of drugs and side effects, Nucleic Acids Res. 44 (2016) D1075–D1079.

[28] M. Kuhn, D. Szklarczyk, A. Franceschini, et al., STITCH 2: an interaction network database for small molecules and proteins, Nucleic Acids Res. 38 (2010) D552–D556.

[29] S. Basith, G. Lee, B. Manavalan, STALLION: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction, Briefings Bioinf. 23 (2022).

[30] M. Jiang, B. Zhao, S. Luo, et al., NeuroPpred-Fuse: an interpretable stacking model for prediction of neuropeptides by fusing sequence information and feature selection methods, Briefings Bioinf. (2021) 22.

[31] Z. Chen, P. Zhao, F. Li, et al., iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences, Bioinformatics 34 (2018) 2499–2502.

[32] H.H. Wu, X.Y. Pan, Y. Yang, et al., Recognizing binding sites of poorly characterized RNA-binding proteins on circular RNAs using attention Siamese network, Briefings Bioinf. (2021) 22.

[33] L.M. Shen, J.Y. Feng, Z. Chen, et al., Self-attention Based Convolutional-LSTM for Android Malware Detection Using Network Traffics Grayscale Image, Applied Intelligence, 2022.

[34] S.Y. Guo, Y. Wang, H. Yuan, et al., TAERT: triple-attentional explainable recommendation with temporal convolutional network, Inf. Sci. 567 (2021) 185–200.

[35] K.M. He, X.Y. Zhang, S.Q. Ren, et al., Deep residual learning for image recognition, in: 2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr), 2016, pp. 770–778.

[36] F. Sun, J. Sun, Q. Zhao, A deep learning method for predicting metabolite-disease associations via graph neural network, Briefings Bioinf. 23 (4) (2022), https://doi.org/10.1093/bib/bbac266.

[37] C.C. Wang, C.D. Han, Q. Zhao, et al., Circular RNAs and complex diseases: from experimental results to computational models, Briefings Bioinf. 22 (6) (2021), https://doi.org/10.1093/bib/bbab286.

[38] W. Liu, Y. Jiang, L. Peng, et al., Inferring gene regulatory networks using the improved markov blanket discovery algorithm, Interdisciplinary Sci. 14 (2022) 168–181.

[39] Q. Tang, F. Nie, J. Kang, et al., mRNALocater: enhance the prediction accuracy of eukaryotic mRNA subcellular localization by using model fusion strategy, Mol. Ther. 29 (2021) 2617–2623.

[40] M.M. Hasan, M.A. Alam, W. Shoombuatong, et al., NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning, Briefings Bioinf. (2021) 22.

[41] S. Basith, M.M. Hasan, G. Lee, et al., Integrative machine learning framework for the identification of cell-specific enhancers from the human genome, Briefings Bioinf. (2021) 22.

[42] Z. Chen, P. Zhao, C. Li, et al., iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization, Nucleic Acids Res. 49 (2021) e60.

[43] X. Wang, F. Li, J. Xu, et al., ASPIRER: a new computational approach for identifying non-classical secreted proteins based on deep learning, Briefings Bioinf. 23 (2022).

[44] J. Hong, Y. Luo, M. Mou, et al., Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery, Briefings Bioinf. 21 (2020) 1825–1836.

[45] Y. Chu, Y. Zhang, Q. Wang, et al., A transformer-based model to predict peptide-HLA class I binding and optimize mutated peptides for vaccine design, Nat. Mach. Intell. 4 (2022) 300–311.

[46] Hendrycks D, Gimpel K. GAUSSIAN ERROR LINEAR UNITS (GELUs), arXiv e-prints 2018.

[47] Kingma DP, Ba JL. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, arXiv e-prints 2017.

[48] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv e-prints 2015.