HAL
open science

DeConDFFuse: Predicting drug-drug interaction using joint deep convolutional transform learning and decision forest fusion framework

Pooja Gupta, Angshul Majumdar, Emilie Chouzenoux, Giovanni Chierchia

To cite this version:

Pooja Gupta, Angshul Majumdar, Emilie Chouzenoux, Giovanni Chierchia. DeConDFFuse: Predicting drug-drug interaction using joint deep convolutional transform learning and decision forest fusion framework. Expert Systems with Applications, 2023, 227, pp.120238:1-22. 10.1016/j.eswa.2023.120238 . hal-04089994

HAL Id: hal-04089994

https://inria.hal.science/hal-04089994v1

Submitted on 5 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DeConDFFuse : Predicting Drug-Drug Interaction using joint Deep Convolutional Transform Learning and Decision Forest fusion framework

Pooja Gupta *
Indraprastha Institute of Information Technology, Delhi, India
poojag@iiitd.ac.in

Angshul Majumdar
Indraprastha Institute of Information Technology, Delhi, India
angshul@iiitd.ac.in

Emilie Chouzenoux
Université Paris-Saclay, CentraleSupélec, Inria, CVN, Gif-sur-Yvette, France
emilie.chouzenoux@inria.fr

Giovanni Chierchia
LIGM, Université Gustave Eiffel, CNRS, ESIEE Paris, Noisy-le-Grand, France
giovanni.chierchia@esiee.fr

*corresponding author: Pooja Gupta, Indraprastha Institute of Information Technology, Delhi, India; Email: poojag@iiitd.ac.in

DeConDFFuse : Predicting Drug-Drug Interaction using joint Deep Convolutional Transform Learning and Decision Forest fusion framework

Pooja Gupta*[a], Angshul Majumdar[a], Emilie Chouzenoux[b], Giovanni Chierchia[c]

[a] *Indraprastha Institute of Information Technology, Delhi, India*
[b] *Université Paris-Saclay, CentraleSupélec, Inria, CVN, Gif-sur-Yvette, France*
[c] *LIGM, Université Gustave Eiffel, CNRS, ESIEE Paris, Noisy-le-Grand, France*

Abstract

In Drug-Drug-Interaction (DDI), the task is to predict the (adverse) effect of administering two drugs simultaneously. Currently, the techniques proposed in this direction are generally based on either shallow learning paradigms like Random Decision Forest (RDF), Logistic Regression (LR), Support Vector Machines (SVM), etc., or deep Convolutional Neural Networks (CNNs). However, specific works combine traditional machine learning (ML) algorithms such as RDF, LR, SVM, and deep learning paradigms such as CNNs in a piecemeal fashion which might not be optimal. Hence, the present work proposes a framework that presents a joint end-to-end solution. We propose a Siamese-like architecture with two processing channels' networks based on deep convolutional transform learning. Common fused representations as well as channel-wise representations are learnt, in addition with the transform across them. The final representation is passed to a decision forest to give final predictions. The proposed method is thus a supervised end-to-end multi-channel fusion framework that (i) learns unique and interpretable filters in contrast with CNNs, and (ii) jointly learns and optimizes decision forest in contrast with state-of-the-art piecemeal approach. We apply this technique to identify DDIs among 1059 drugs from the DrugBank database showing superiority of our method compared to the state-of-the-art(s).

Keywords: information fusion; convolution; drug-drug interaction; multi-channel; transform learning ; decision forest;

1. Introduction

Drug-Drug interactions (DDIs) are the adverse changes or effects or reactions of one drug due to the recent concurrent use of another drug(s). For example, the drug Ceftriaxone should be avoided in children less than 28 days old if they are receiving or expected to receive IV calcium-containing products. Indeed, it might lead to neonatal deaths resulting from crystalline deposits in the lungs and kidneys, as reported in (Sandritter et al.). Such reaction from DDIs is known as adverse drug reactions (ADRs). ADRs are responsible for the threat to a person's life and inadvertently increase overall healthcare costs.

According to the studies (Allison, 2012; Bouvy et al., 2015), ADRs contribute to more than 20% of clinical trial failures and are considered the highest load in the modern drug

**corresponding author email: poojag@iiitd.ac.in*

discovery process. Serious ADRs can cause severe disability and even death in patients. Also, from study (Bouvy et al., 2015), it is observed that approximately 3.6% of all hospital admissions are caused by ADRs in Europe. Up to 10% of patients in European hospitals experience an ADR among those patients. Similarly, it has been estimated that more than 2 million severe ADRs occur in hospitalized patients each year in the United States. This results in more than $100,000$ deaths (Giacomini et al., 2007; Lounkine et al., 2012). From a financial perspective, the annual financial cost of drug-related morbidity in the United States (US) was estimated at $528.4 billion in 2016, equivalent to 16% of total US healthcare expenditures that year (Jonathan H. Watanabe, 2018).

It, thus, becomes pertinent to identify in an exhaustive manner, the DDIs that could cause ADRs. This might not avoid all unanticipated drug interactions but it can help lower the drug development costs and optimize the drug design process (Zhang et al., 2021). Initially and primarily now also, the techniques for DDIs identification are based on clinical trials and experimentation conducted within a living organism (in vivo) or outside it (in vitro) (Duke et al., 2012). While the ideal case would be to identify all the possible interactions during clinical trials, most of these are often practically determined after the approval of drugs for clinical use. The reason is that the sample size and duration of pre-market trials are limited (Whitebread et al., 2005). Another problem with the clinical trials is that the trial subjects are at risk for potential adverse effects.

The other strategy relies on computational approaches. This involves predicting DDIs from some drug features via algorithmic approaches such as machine learning, deep learning, etc. Many significant studies are suggestive of such techniques. The features (i.e., the properties) of the drugs are extracted, such as their side effects, their target proteins, enzymes, chemical structures, etc. Then, off-the-shelf supervised machine learning algorithms such as Decision Trees (DT), Naive Bayes (NB), Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GBT), K-nearest neighbors (KNN), etc. are applied to finally predict probable DDIs given these features (Abdelaziz et al., 2017). Exploiting and finding similarities between drug features can be useful in that context (Zhang et al., 2019). However, this method can deal with only few (typically up to four) drug features and requires learning a high number of learning parameters that might lead to high training costs. Recently, some works have represented the DDI problem as a matrix completion problem (Zhang et al., 2018; Mongia et al., 2020). Nevertheless, this approach too can involve only a limited set of drug features while a high number of various diverse features are usually available for a drug.

Deep Learning (DL) approaches are currently used in almost every area of problem-solving. In the context of DDI, DL allows the automated learning of features from raw data and provides an end-to-end solution. Approaches like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) such as Long-Short Term Memory (LSTM) have been used to predict DDIs. In the study (Huang et al., 2017), the authors have investigated two-stages framework using feature-based binary classifier first and then uses LSTM network.

Researchers have used Knowledge Graphs (KGs) to represent drugs' information in this respect. They usually combine the diverse information available about drugs across many different sources in the form of Knowledge Graphs. After that, they apply the DL techniques on them (Rezaul Karim et al., 2019) like CNNs or recurrent neural networks like LSTM. Still, all of these aforementioned techniques suffer from the typical black-box issue reported in DL, with no interpretability nor uniqueness guarantees on the learnt inner representations.

In this work, we propose a novel model to solve the DDI problem in a supervised fashion. We propose to adopt a representation learning paradigm to learn diverse and interpretable features giving a common representation for the two drugs. We then introduce an end-to-end multi-channel framework that fuses the information about the drugs that are present in a drug pair, and finally gives our DDI predictions from a decision forest. The task at hand is to predict DDIs that could be of two types, namely 1 (known-to-interact) and 0 (either unknown or known-not-to-interact) interactions. Hence, it is a binary classification task. The whole pipeline is jointly and globally optimized to further guide learnt representations in the same and optimal decision. Its performance is compared over the state-of-the-arts on a DDI dataset. The rest of the paper is organized as follows. We first perform a bibliographical study in Section 2. We then discuss the proposed method and dataset in Section 3. In Section 4, we proceed with the experimental setup, followed by the results and some discussion in Section 5. Finally, we conclude our work in Section 6.

2. Related work and positioning

DDIs identification is considered as a non-trivial problem from the research perspective to be solved in the pharmacology discipline. In literature, many different computational strategies have been investigated that we discuss here below. Several families of methods can be identified, relying either on statistical machine learning models, graph models, deep learning models and matrix factorization models.

2.1. Statistical Machine Learning based frameworks

The work (Sridhar et al., 2016) proposes similarity-based models that compute similarity scores between drug features like chemical structures, side-effects, targets, pathways, etc., and thereafter performs a probabilistic inference of the DDIs. Researchers have explored Bayesian learning models (Yu et al., 2008) under statistical learning paradigms. Another work (Zhang et al., 2019) uses sparse feature learning ensemble method with linear regularization utilizing four drug features - chemical substructures, targets, enzymes and pathways. In (Ferdousi et al., 2017), the work utilizes the drug similarity function between various drug features such as the chemical structure, enzymes, targets, pathways, etc., to compute DDIs. In (Dang et al., 2021), ML algorithms like NB, DT, RF, LR, and XGBoost were used with cross-validation with input as SMILE values and interaction features based on CYP450 group. In (Huang et al., 2017), the two-stages framework is proposed using SVM in first stage as feature based binary classifier and then a RNN based bidirectional LSTM network.

All the aforementioned studies utilize statistical machine learning methods whose performance might depend highly on the quality of features used; thus, it becomes pertinent to explore multiple features than restrict them to some set. Furthermore, overfitting stays a significant issue with these techniques due to their restrictive non-linear mapping and fitting capability.

2.2. Matrix Factorization and multi-modal techniques

Let us also mention recent studies that present matrix factorization as the solution to predict DDIs (Zhang et al., 2018; Mongia et al., 2020). Here, the input is the DDI matrix and the similarity scores between the drugs. The pair for which the DDI is to be predicted

is treated as a missing value; hence, it is imputed using the inputted similarity scores. Then some works use the Triple Matrix Factorization also (Shi et al., 2018, 2017). Matrix Factorization based algorithms may have problems if the values are not independent. Additionally, this technique is likely to fail as it assumes missing values at random which is not the case in practice. Some researchers have even proposed multi-modal techniques to predict DDIs. Although there are few, we briefly introduce a few of these studies here. The study that used this technique learned the unified drug representations from multiple drug feature networks simultaneously using multi-modal deep auto-encoders. Then, they applied four operators on the learned drug embeddings to represent drug-drug pairs, and finally, they use a RF classifier to train models for predicting DDIs (Zhang et al., 2020).

2.3. Graph-based frameworks

Graph-based embedding techniques are also gaining momentum in DDIs prediction. With the advent of the availability of biomedical data, researchers are moving toward KGs to populate and complete the available biomedical information. It is done with the help of the large structured databases and texts available publicly (Celebi et al., 2019). For example, the Bio2RDF project has made 35 life sciences datasets available as Linked Open Data (LOD) in RDF. In this, similar entities are mapped in different KGs, resulting in large heterogeneous biomedical graphs containing drug-related facts. Some works have used the combination of DDI matrix and KG followed by the application of ML algorithms (Celebi et al., 2019).

2.4. Deep Learning based frameworks

Deep learning is another effective modeling technique that is extensively used in solving most real-world problems these days. It has emerged to be helpful in the said DDI prediction as well. In (Yu et al., 2022), the proposed framework integrates the multi-relational and the relation-aware network structure representations. Finally, the integrated representations via concatenation is passed through neural network to get the final DDI predictions. The study (Sahu & Anand, 2018) proposes attention-based RNNs - LSTM for DDI prediction. In (Liu et al., 2022), the work utilizes deep Neural Networks based on attention technique for predicting DDIs with features from multiple networks are learnt using graph embedding techniques. In (Rohani & Eslahchi, 2019), multiple drug similarities based on drug substructure, target, side effect, off-label side effect, pathway, transporter, and indication data are calculated. Then, it uses a neural network for interaction prediction.

In another study (Chen et al., 2019), the authors explore the molecular graphs formed from SMILE inputs and learn the graph representations using Siamese Graph Convolutional Network (GCN) and further pass these representations to the neural networks for the final predictions. Some works have even integrated the CNNS and RNNS for the same. For example, (Liu et al., 2020) integrates CNNs, RNNs and mixture density networks. Another work (Wu et al., 2020) combines RNN-based stacked Gated Recurrent Units (GRU) with CNNS for the prediction task.

Further studies are combining KGs and DL to predict DDIs. In the study (Rezaul Karim et al., 2019), the DDI matrix and KG form the input to the DL network that has CNN and LSTM. KG is input to the network in the form of learned embeddings like ComplEx, TransE, RDF2Vec, etc. This work (Park et al., 2020) uses a Graph CNN and additionally applies attention based strategy to prune the irrelevant information and keep the significant parts

only. Another work suggests using a Neural Network instead of CNN that uses the same input as in the preceding study mentioned, except that it is based on previously established Graph Neural Networks (GNN) (Hamilton et al., 2017). Also, it focuses on neighborhood sampling and aggregates entities and their neighborhood representation into a single vector in 3 different ways (Lin et al., 2020). Another work (Chen et al., 2021) proposes the Deep CNN network that learns the cross channel features from two inputs - KG and drug molecular features graph.

From the above mentioned studies related to DDI prediction and otherwise also, CNNs have been regarded as one of the most frequently used paradigms for solving problems. However, there are certain issues that persist with CNNs. There is no surety of learning of distinctive filters and hence there are chances of redundancy in the learned features / representations. Another issue that is encountered with CNNs is dead neuron problem that becomes big in nature if every neuron in a specific hidden layer is dead, it cuts the gradient to the previous layer resulting in zero gradients to the layers behind it. Thus, the weights would not be updated and the learning will be improper.

Also, the current proposed paradigms are usually such that these learn features from CNNs and then explicitly employ some other classifier to perform the classification task, i.e. piecemeal approach instead of joint training that might have chance of loosing important information. With other techniques as well like similarity-based, network-based, graph-based, etc., we do not obtain guarantee of the distinctive learning of features. Thus, from this discussion, these issues lead to the scope for developing supervised frameworks that can combinely tackle them.

2.5. Our contribution

Our work relies on a supervised learning strategy formulating the problem as a binary classification one, which makes our approach related to the mentioned DL methods. However, the proposed pipeline involves representation learning steps, namely convolutional transform learning, that aims at gaining interpretability and stability of the decision process. In particular, our specific learning strategy enhances unique features/representations that help in better prediction. The features/representations are also optimized and learned in a direction given by a decision forest predictor with the goal to reach better predictions for the test data. Thus, this work proposes an end-to-end solution that learns parameters for both deep CTL and decision forest at the exception of the RF probability distribution. The latter is computed based on the CTL and decision forest learnt models. The end-to-end training relies efficiently on stochastic gradient updates via automatic differentiation. The details of our pipeline and our training strategy are discussed in the next Section 3.

Also, the works discussed above utilize different feature sets for drugs that may not be readily available for some small molecule drugs and hence, limit them to predict interactions for those with the other drugs, for example, (Zhang et al., 2019). Thus, another benefit with our approach is that we have used Bioactivity descriptors as features for the drugs in our dataset, generated via Signaturizer tool (Bertoni et al., 2021). The pre-trained Siamese Neural Network of the tool helps to infer 25 different types of bioactivity descriptors for the drugs that have little or no information. With our approach, we initially need smile values which we may not get for all the drugs but still will be able to process more drugs when compared with feature sets that reduces the number of drugs significantly. For example,

the features used in the study (Zhang et al., 2019) that restrict the number of the drugs considered and compels to remove them from the dataset due to non-availability of features.

3. Data and Methodology

This section will first briefly describe the dataset used, followed by the proposed technique for inference and training.

3.1. Dataset Description

We use DDI data from Stanford's Biosnap dataset [2], which contains a network of 1514 DrugBank drugs representing nodes and 48514 drug-drug interactions representing edges. This network of interactions between drugs is approved by the U.S. Food and Drug Administration. We have assumed all other interactions apart from approved interactions as either known-not-to-interact or unknown. Here, we represent the known-to-interact interactions by 1 and the others by 0 numerically. We first determined the SMILE values of the drugs using compound IDs taken from the dataset using DrugBank.ca. Since the SMILE values are not available for all the drugs (retrieved using DrugBank IDs), thus, the number of the drugs in the dataset was reduced to 1368 and, accordingly, the number of interactions. Further, we have processed only the drugs that have at least 10 known-to-interact interactions with other drugs. So, there are 1059 drugs and their respective interactions.

Thereafter, we extract the bioactivity descriptors via the Signaturizer tool (Bertoni et al., 2021) using the determined smile value of each drug. This tool provides bioactivity descriptors that encode the physicochemical and structural properties of small molecule drugs covering all the drugs present in Chemical Checker (CC). The latter has further covered the source databases - DrugBank.ca and ChEMBl. It has a pre-trained Siamese Neural Network via which inputting a smile value for drug, 25 different types of bioactivity descriptors can be inferred for the drugs that have little or no information. The descriptors are fixed-length normalized vectors of size 128. There are broadly five categories of bioactivity descriptors labeled as A to E (A: Chemistry, B: Targets, C: Networks, D: Cells, and E: Clinics). Each has five sub-categories marked as A1 to A5, for example, thus 25 different descriptors. We have taken descriptors from A and B broad categories representing a drug's Chemistry and Targets, respectively. Further, we choose A1 and A2 sub-categories from A, representing 2D and 3D fingerprints, and the B1 sub-category from B, representing the mechanism of action of a drug. Since we have chosen 3 types of bioactivity descriptors out of 25, each having 128 fixed-sized vectors, therefore, we have a total $384(128 \times 3)$ features for every drug. Thus, our dataset comprises 1059 unique drugs with 384 bioactivity descriptors/features for each drug and corresponding interactions.

3.2. Proposed Framework

In this section, we discuss our proposed work. We present a fusion framework that combines the benefits of our recently established multi-channel, unsupervised, fusion-based

[2]https://snap.stanford.edu/biodata/datasets/10001/10001-ChCh-Miner.html#: :text=Dataset %20information&text=Drug%2Ddrug%20interactions%20occur%20when,such%20as%20adverse %20drug%20reactions

representation learning framework - DeConFuse (Gupta et al., 2020) and jointly optimizes a decision forest with binary decision, that gives the final DDI Predictions. Such a solution has been successfully used before in Deep Neural Decision Forest (DNDF) framework (Kontschieder et al., 2015). Let us mention that DeConFuse architecture is unsupervised. We want to propose a supervised version of this architecture. We previously established a supervised version of this framework namely SuperDeConfuse (Gupta et al., 2021). However, the supervision in SuperDeConFuse was incorporated by using cross entropy loss in the optimization objective and using softmax in the end of its architecture. Here, our goal is to guide the supervision through a random decision forest. The proposed solution does not utilize features/representations from CNN but instead from the DeConFuse network based on deep CTL, through linear transform learning. The advantage that latter offers is that it promotes unique filters/transforms which is not guaranteed with CNNs. Such advantage helps in learning diverse and interpretable representations. These representations are further guided by the predictions from decision forest whose parameters are jointly optimized. The representations learnt are useful as these help decision forest to correctly identify many known-to-interact (1) DDIs as also corroborated from the experimental results discussed in the section 5 later. We briefly discuss both frameworks and, finally, mention the details of the combined fusion framework.

3.2.1. DeConFuse

Let us start introducing Deep CTL architecture for representation learning. It stacks multiple convolutional layers on top of each other to generate the features, as shown in Figure 1.

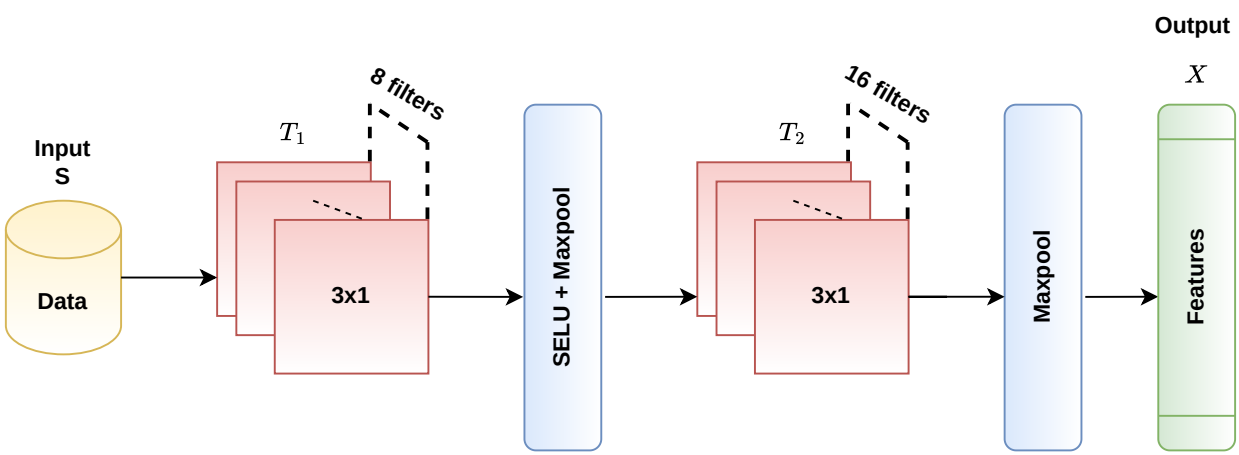


Figure 1: Deep CTL architecture. The illustration is given for $L = 2$ layers, with the first layer T_1 composed of $M_1 = 8$ filters of size 3×1, and the second layer composed of $M_2 = 16$ filters of size 3×1.

Here $T_\ell, \ell \in \{1, \ldots, L\}$ filters are convoluted with input sample S. The outputs X are the learned features corresponding to the convoluted output. Optimal filters aim at reducing the quadratic loss

$$\widehat{F}_{\text{conv}}(T_1, \ldots, T_L, X \mid S) = \frac{1}{2}\|T_L * \phi_{L-1}(T_{L-1} * \ldots \phi_1(T_1 * S)) - X\|_F^2, \tag{1}$$

where ϕ is the activation function (e.g., SELU), and L represents the number of CTL layers we apply, which in our case is $L = 2$. However, this objective alone is not enough for proper learning, as it can give trivial solution (e.g., X and T_ℓ all equal to 0). To avoid this situation, regularization is added, leading to the objective function

$$F_{\text{conv}}(T_1, \ldots, T_L, X \mid S) = \frac{1}{2}\|T_L * \phi_{L-1}(T_{L-1} * \ldots \phi_1(T_1 * S)) - X\|_F^2 + \iota_+(X) + \sum_{\ell=1}^{L} (\mu||T_\ell||_F^2 - \lambda \log\det(T_\ell)). \tag{2}$$

Here, the term $\iota_+(X)$ is a non-negativity constraint on X (equals to 0 for positive valued X, $+\infty$ otherwise). The regularization term "$\mu \|\cdot\|_F^2 - \lambda \log\det$" ensures that the non-zero and unique filters are learnt (which is not guaranteed in CNN). We learn here all the variables in an end-to-end fashion. Next, we extend the deep CTL to a fusion network where we have two separate Deep CTL networks/channels ($C = 2$) for each drug in a drug pair that give features $X = (X^{(c)})_{1 \leq c \leq C}$ and fused together to give the common representation Z. The latter representation is learnt via linear transforms (i.e. not convolutional) as learnt in original Transform Learning (TL) technique (Ravishankar & Bresler, 2013). This part of the architecture is learnt so as to reduce the fusion loss

$$F_{\text{fusion}}(\widetilde{T}, Z, X) = \frac{1}{2}\left\|Z - \sum_{c=1}^{C} \text{flat}(X^{(c)})\widetilde{T}_c\right\|_F^2 + \iota_+(Z) + \sum_{c=1}^{C} \left(\mu\|\widetilde{T}_c\|_F^2 - \lambda \log\det(\widetilde{T}_c)\right) \tag{3}$$

where the operator "flat" transforms $X^{(c)}$ into a matrix where each row contains the "flattened" features of a sample. In a nutshell, DeConFuse has two parts, (i) Deep Convolutional part and (ii) Fusion part. The learning procedure aims at solving the joint optimization problem given as:

$$\underset{T, X, \widetilde{T}, Z}{\text{minimize}} \;\; F_{\text{fusion}}(\widetilde{T}, Z, X) + \sum_{c=1}^{C} F_{\text{conv}}(T_1^{(c)}, \ldots, T_L^{(c)}, X^{(c)} \mid S^{(c)}). \tag{4}$$

The complete architecture of DeConfuse is given in Figure 2. There are two notable advantages of the DeConFuse approach. Firstly, we rely on automatic differentiation (Paszke et al., 2017) and stochastic gradient approximations to efficiently solve Problem (4). Secondly, we are not limited to ReLU activation in (2), but rather we can use more advanced ones, such as SELU (Klambauer et al., 2017). It is beneficial for the performance, as shown by our numerical results.

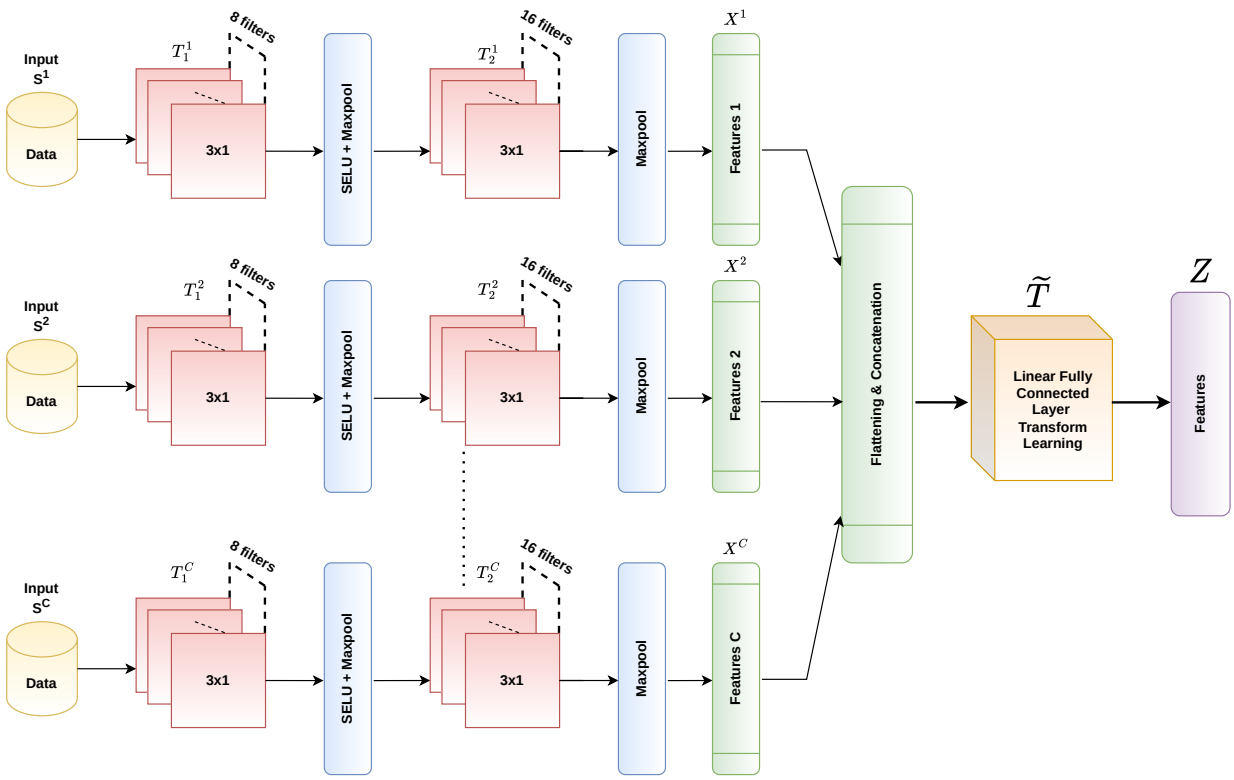


Figure 2: DeConFuse architecture. This is a general architecture representation where C represents the number of DeepCTL networks/channels.

3.2.2. DNDF Framework

We now introduce the DNDF framework from (Kontschieder et al., 2015), that will be the last brick of our DDI pipeline. It is different from conventional deep neural networks as it outputs the final predictions from the decision forest, and their split (decision nodes) and leaf (prediction) nodes' parameters are jointly and globally optimized. The technique is stochastic, differentiable, and, thus, gives a backpropagation compatible version of decision trees that guides the representation learning in lower layers of deep CNNs. This reduces the uncertainty on routing decisions of a sample taken at the split nodes, such that the globally defined loss function is minimized. For the leaf nodes, the optimal predictions are achieved by minimizing the convex objective function, which does not require step size selection. We further explain the objective function briefly.

Decision Trees with Stochastic Routing

Consider a classification problem with input space χ and finite output space Y. A decision tree consists of decision (or split) nodes and prediction (or leaf) nodes. Decision nodes, let's say, indexed by N are internal nodes of the tree, and prediction nodes are indexed by $\mathcal{L}$, i.e., terminal/leaf nodes of the tree. Each prediction node $\ell \in \mathcal{L}$ is associated with a probability distribution $\pi_\ell = (\pi_{\ell_y})_{y \in Y}$. Each decision node $n \in N$ is assigned a decision function $d_n(\cdot;\theta) : \chi \to [0,1]$ parameterized by θ, which routes the samples along the tree branches. A sample $x \in \chi$ when reaches a decision node n, it will be directed either to the left or right sub-tree based on the output of the function $d_n(x;\theta)$. Here, it is a probabilistic

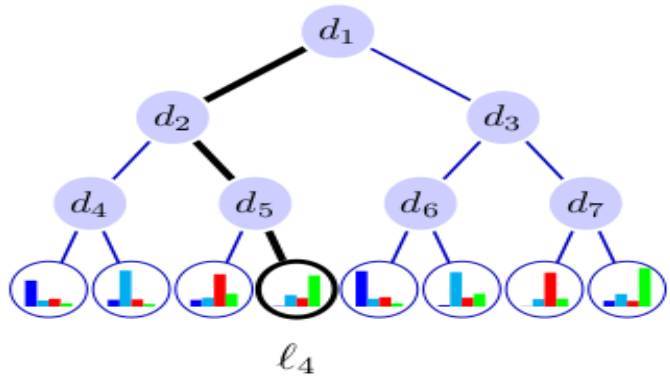


Figure 3: Each node $n \in N$ of the tree performs routing decisions via function $d_n(\cdot)$. The black path shows an exemplary routing of a sample x along a tree to reach leaf ℓ_4, which has probability $\mu_{\ell_4} = d_1(x)\bar{d}_2(x)\bar{d}_5(x)$. *Image taken from (Kontschieder et al., 2015).*

routing where the routing direction is the output of a Bernoulli random variable with mean $d_n(x;\theta)$. As the sample ends in a leaf node ℓ, the related tree prediction is given by the class-label distribution $\pi_\ell = (\pi_{\ell_y})_{y \in Y}$. In the case of stochastic routings, the leaf predictions will be averaged by the probability of reaching the leaf. Thus, the final prediction for a sample x from tree D with decision nodes parameterized by θ is given as:

$$(\forall y \in Y) \qquad \mathbb{P}_D[\, y \mid x, \theta, \pi] \, = \sum_{\ell \in \mathcal{L}} \pi_{\ell_y} \, \mu_\ell(x \mid \theta) \tag{5}$$

where $\pi = (\pi_\ell)_{\ell \in \mathcal{L}}$. Here above, $\mu_\ell(x \mid \theta)$ is regarded as the routing function providing the probability that sample x will reach leaf ℓ. Note that $\Sigma_\ell \mu_\ell(x \mid \theta) = 1$ for any $x \in \chi$.

For an explicit form for the routing function, the following binary relations that depend on the tree's structure are given as: $\ell \swarrow n$, which is true if ℓ belongs to the left sub-tree of node n, and $n \searrow \ell$, which is true if ℓ belongs to the right sub-tree of node n. Hence, these relations can be exploited to express μ_ℓ as:

$$\mu_\ell(x \mid \theta) = \prod_{n \in N} d_n(x;\theta)^{\mathbb{1}_{\ell \swarrow n}} \; \bar{d}_n(x;\theta) \tag{6}$$

where $\bar{d}_n(x;\theta) = 1 - d_n(x;\theta)$, and $\mathbb{1}_P$ is an indicator function conditioned on the argument P. Although the product in (6) runs over all nodes, however, only decision nodes along the path from the root node to the leaf ℓ contribute to μ_ℓ, because for all other nodes $\mathbb{1}_{\ell \swarrow n}$ and $\mathbb{1}_{n \searrow \ell}$ will be both 0 (with the assumption $0^0 = 1$). See Figure 3.

Decision functions deliver a stochastic routing with decision functions defined as follows:

$$d_n(x;\theta) = \sigma(f_n(x;\theta)), \tag{7}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function, and $f_n(\cdot;\theta) : \chi \to \mathbb{R}$ is a real-valued function depending on the sample and the parameterization θ.

A forest is an ensemble of decision trees $\mathcal{F} = \{D_1, \ldots, D_k\}$, which delivers a prediction for a sample x by averaging the output of each tree, i.e.

$$(\forall y \in Y) \qquad \mathbb{P}_{\mathcal{F}}[y \mid x, \theta, \pi] = \frac{1}{k} \sum_{h=1}^{k} \mathbb{P}_{D_h}[y \mid x, \theta, \pi]. \tag{8}$$

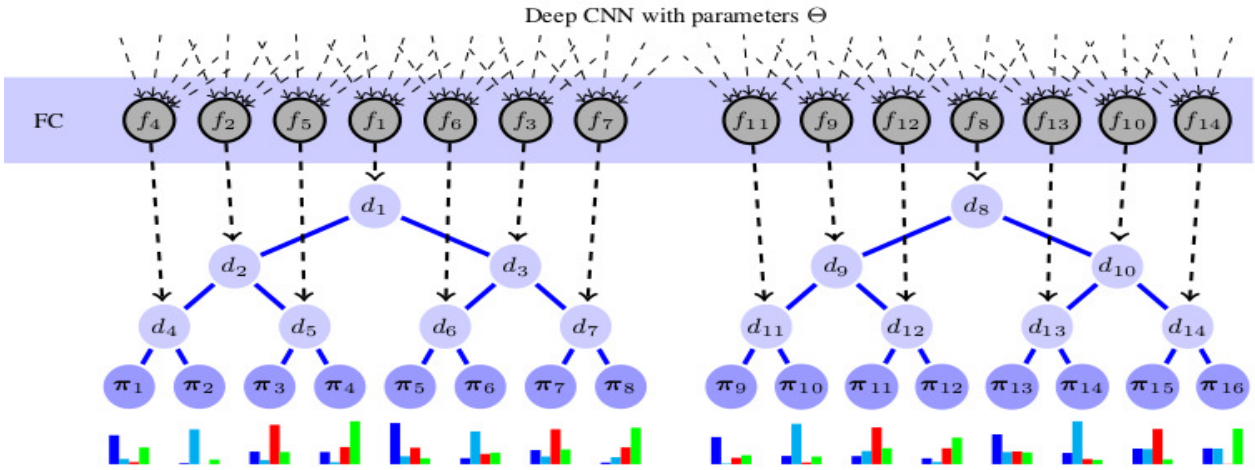


Figure 4: illustration of how to implement a deep neural decision forest (DNDF). Top: Deep CNN with a variable number of layers, subsumed via parameters θ. FC block: Fully Connected layer used to provide functions $f_n(\cdot;\theta)$, described in Equ. 7. Each output of f_n is brought in correspondence with a split node in a tree, eventually producing the routing (split) decisions $d_n(x) = \sigma(f_n(x))$. The order of the assignments of output units to decision nodes can be arbitrary (the one shown allows a simple visualization). The circles at the bottom correspond to leaf nodes, holding probability distributions π_ℓ. *Image taken from (Kontschieder et al., 2015).*

Learning Trees by Back-Propagation

Learning a decision tree, for which the model is explained in the previous sections, requires estimating both the decision node parameterizations θ and the leaf predictions π. The parameters θ are estimated using the Minimum Empirical Risk principle with respect to a given data set $\mathcal{T} \subset \chi \times Y$ under the log-loss (also known as the cross-entropy loss), i.e., minimizers of the following risk term are searched:

$$F_{\text{tree}}(\theta;\pi;\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} -\log(\mathbb{P}_D[y \mid x,\theta,\pi]) \tag{9}$$

The forest is learned by considering the ensemble of trees $\mathcal{F}$, where all trees can possibly share the parameters in θ. Still, each tree can have a different structure with a different set of decision functions and independent leaf predictions π. The illustration of the forest of decision trees taking the parameters θ and computing routing decisions and prediction nodes probabilities can be referred to from Figure 4. Thus, for the forest, empirical risk is minimized as:

$$F_{\text{forest}}(\theta;\pi;\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} -\log(\mathbb{P}_{\mathcal{F}}[y \mid x,\theta,\pi]). \tag{10}$$

A two-step optimization strategy is followed to minimize the above function, with alternate updates of θ and π. Interested Readers can further know about the detailed updates/learning mechanism for the parameters θ and π from DNDF (Kontschieder et al., 2015).

3.2.3. Combined Proposed Framework DeConDFFuse - DeConFuse and Decision Forest

We propose to combine the frameworks explained in the previous sections 3.2.1 and 3.2.2. Specifically, instead of utilizing the features from a CNN network, we propose to inherit the

representations learned from the DeConFuse network to peruse them in our decision forest, i.e., the decision forest is jointly trained and optimized within the DeConFuse network. The DeConFuse network learns channel-wise representations corresponding to each drug in a drug pair, that is $X^{(c)}$ with $c \in \{1, 2\}$, and finally learns common representation Z from $X^{(c)}$ where the fusion takes place. Here, we do not have a positivity constraint on Z and only on channel-wise representations $X^{(c)}$.

The representation Z is passed to the DF, where it applies the features mask, i.e., randomly selects the features from the representation that will participate in the decision tree's routing process that sends those selected features to the linear fully connected layer parameterized by θ, i.e., given by the function $f_n(x; \theta_n) = \theta_n^\top x$. The number of features involved is given by feature ratio. Thereafter, the sigmoid activation is applied as given in Eq. 7. Then the routing function is computed, and the prediction probabilities are calculated. Thus, the prediction probabilities having a probability for each class for each tree is likewise obtained. Finally, the probabilities from all the trees of the Forest $\mathcal{F}$ are averaged to get the outcome probability for each of the classes 0 and 1 in our case. The negative log-likelihood loss is computed and back-propagated, which guides the representation learning of the DeConFuse framework and learning of the parameters θ. The objective function for this framework that combines the idea of DeConFuse and DF can be deduced from 4 and 10:

$$\underset{T,X,\widetilde{T},Z,\theta,\pi}{\text{minimize}} \quad \underbrace{F_{\text{fusion}}(\widetilde{T}, Z, X) + \sum_{c=1}^{C} F_{\text{conv}}(T_1^{(c)}, \ldots, T_L^{(c)}, X^{(c)} \mid S^{(c)}) + F_{\text{forest}}(\theta; \pi; \mathcal{T}_Z)}_{J(T,X,\widetilde{T},Z,\theta,\pi)}. \tag{11}$$

Hereabove, the dataset $\mathcal{T}_Z$ is built with the learned features Z and the known labels. Note that there is no positivity constraint anymore on the learned representations Z.

4. Experimental Setup

We conduct experiments on drug-drug interaction dataset comprising DDI matrix and bioactivity descriptors/feature vectors for each drug as explained in section 3.1. We divide the DDI matrix dataset into training and testing datasets. We have kept all drugs in the training data so that there are 95 samples per drug. Further, out of 95 samples, there are 60% of 1 interactions for each drug (not exceeding half of the 95, i.e., min (60% of 1 interactions, 95//2)), and the remaining are the samples from 0 interactions. The remaining samples of 0 and 1 interactions per drug are kept in testing data. All the training and test data samples from each interaction category per drug are selected randomly. Also, only one pair of interactions are kept from either the upper triangle or the lower triangle of the DDI matrix. Thus, each training and testing sample is the drug pair and its corresponding interaction value, which we call a label. Approximately there are 1L training samples and 4L testing samples.

We pass the input as a drug pair as a sample during training. For each drug in a pair, we give the 1D feature vectors i.e. bioactivity descriptors to the individual channel/network based on deep CTL, where $L = 2$ represents the number of CTL layers. Thus, the input

S gathers the bioactivity descriptors/1D feature vectors of size 384 for each channel corresponding to each drug. Since there are 1D feature vectors for each drug in the drug pair, thus, we have 1D convolutions in each deep CTL network. The two networks' learned features/representations $X^{(c)}$ are flattened and concatenated. Then we pass these features to the linear Transform learning layer that acts as a fully connected layer where we learn transform $\widetilde{T}$ and common representation Z. Further, we send the learned representation Z selectively by applying the feature mask to the decision forest. The final predictions are output by averaging the predictions from each tree in the decision forest.

The complete architecture is shown in Figure 5 and all the architectural and hyperparameter details are given in Table 1. In the set of hyperparameters, the atom ratio signifies the number of features to be kept in the representation Z; and the feature ratio signifies the randomly selected number of features from the representation Z that will participate in the routing decision function of each tree parameterized by θ.

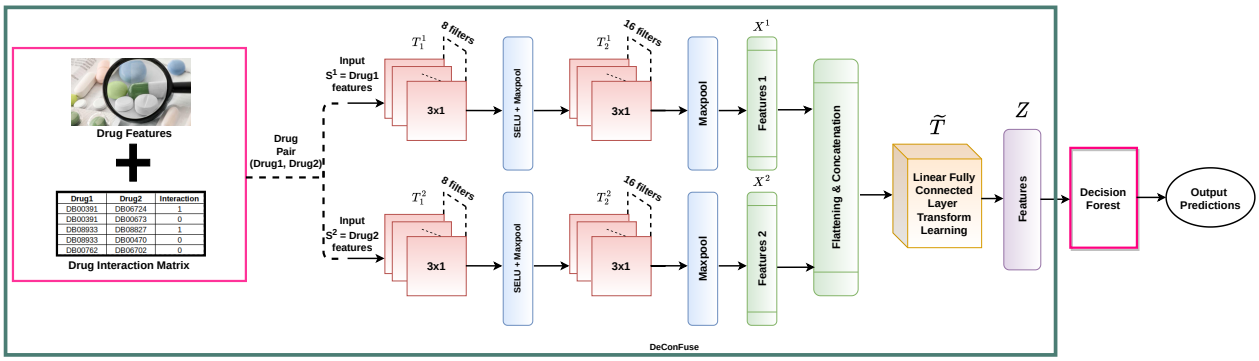


Figure 5: DDI prediction using combined DeConFuse and decision forest architecture- DeConDFFuse. Here $C = 2$ number of networks/channels via each of which a drug in the drug pair is passed along with its bioactivity descriptors/ features vector respectively.

We have compared our results with three state-of-the-arts/benchmark techniques, namely:

- **KGNN**: This technique is used to build the Knowledge Graph (KG) and pass the DDI matrix and KG to the Graph Neural Network (GNN). It focuses on neighborhood sampling and aggregates entities and their neighborhood representation into a single vector in 3 ways - sum, concat, and neighbor (Lin et al., 2020).

- **Conv-LSTM**: This technique uses the DDI matrix and KG in the form of the input to the DL network that has a CNN and LSTM. KG is input to the network in the form of learned embeddings like ComplEx, TransE, RDF2Vec, etc. (Rezaul Karim et al., 2019). We have compared against the embedding that has given the best results in this work, i.e., ComplEx embedding.

- **Graph Embedding DDI**: This technique uses KG and DDI matrix as input but experiments with many different types of embedding techniques. Then each of these embeddings, one by one, are passed to machine learning techniques like Random Decision Forest (RF), Gaussian Naive Bayes (GNB), and Logistic Regression (LR) (Celebi et al., 2019). Here also, we have used the embedding type Skip Gram, which gives the best results in their study.

Table 1: DDI Prediction DeConDFFuse Architecture Details

Parameter	Value
Layer Wise Hyperparameters	
Layer1 - Convolution (CTL)	(1,16,3,1,1)
Maxpool	(2,2)
Layer2 - Convolution (CTL)	(1,32,3,1,1)
Maxpool	(2,2)
atom ratio	0.75
Decision Forest Hyperparameters	
#Trees	90
tree depth	10
feature ratio	0.5
Other Model Hyperparameters	
Epochs	75
Learning Rate	0.01
μ	1e-05
λ	0.0001
batch size	4096
weight decay	1e-05
Optimizer Hyperparameters	
Optimizer Used	Adam
Ams grad	True
Learning rate	0.01
betas	(0.9, 0.999)
eps	1e-08

(in˙planes, out˙planes, kernel_size, stride, padding)
(kernel_size, stride)

For all three benchmarks - KGNN, KG Conv-LSTM, and Graph Embedding DDI- we have used the same DrugBank IDs as present in our training and testing samples. Since these methods rely on KGs, we did not use the bioactivity descriptors/features but recreated KG and embeddings for our dataset, for running these benchmarks.

5. Results and Analysis

The prediction results are evaluated using the classification metrics - AUC_ROC, F1 Score, Precision, Recall, and Accuracy. We have computed all the metrics except Accuracy as weighted metrics since there is a huge class imbalance between 0 and 1 labels. The following Table 2 contains the values of the said evaluation metrics:

We have also computed the confusion matrices (in percentages) for each of the methods. They are displayed in Figure 6.

From Table 2, it is seen that benchmark Graph DDI gives the best values in terms of Accuracy, F1 Score and Recall, and our method for Precision, AUC ROC, and AUPRC.

Table 2: **DDI Prediction Results**

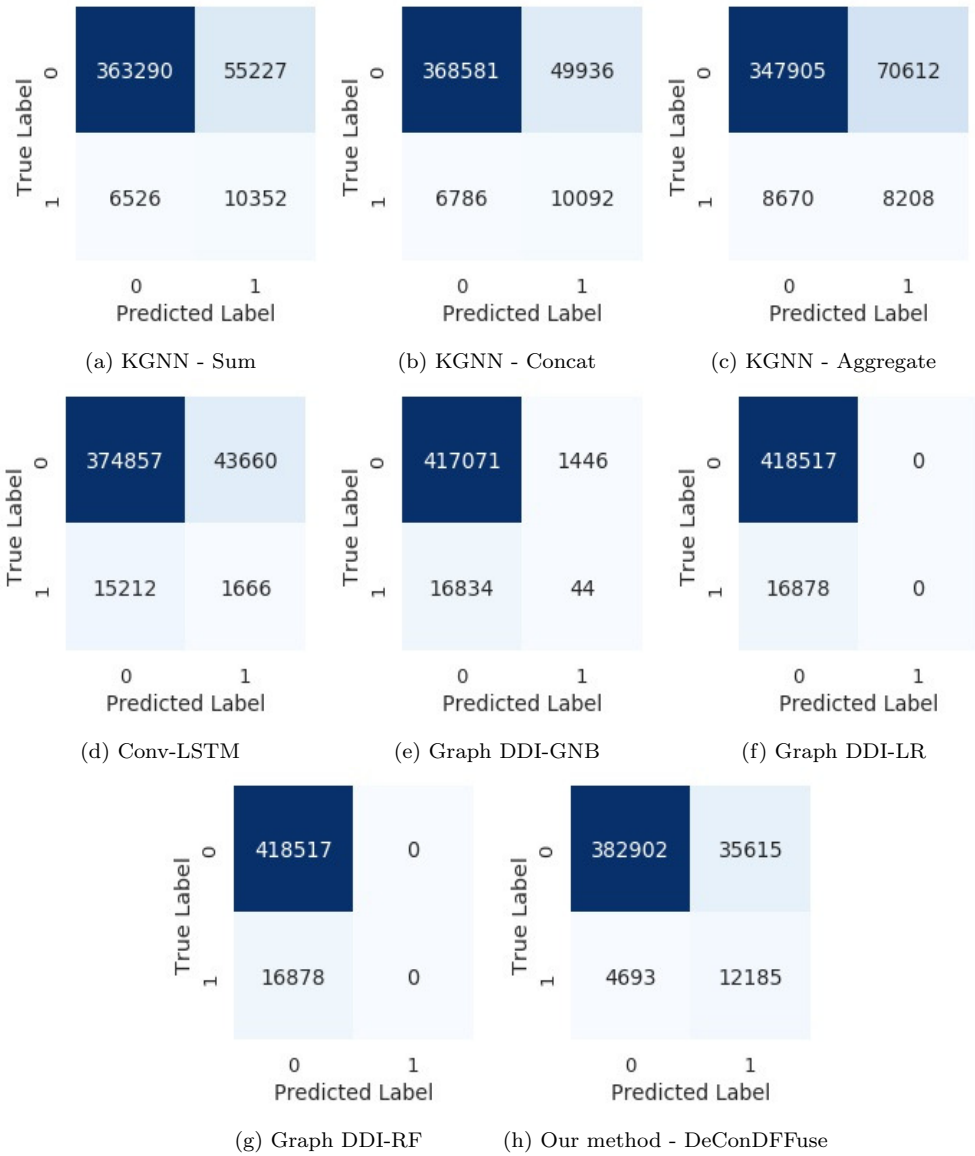
Method	Sub Method	Accuracy	F1	Precision	Recall	AUC ROC	AUPRC
KGNN	**Sum**	85.8168	89.5672	95.0392	85.8168	82.6508	18.6945
	Concat	86.9723	90.2730	95.0375	86.9723	83.5235	19.8427
	Neighbor	81.7908	86.9563	94.1900	81.7908	74.379	10.7655
Conv-LSTM ComplEx	-	86.4785	89.3325	92.5174	86.4785	49.597	3.8164
Graph DDI Skip Gram	**GNB**	95.8015	94.0807	92.5087	95.8015	50.1899	3.8546
	LR	**96.1235**	**94.2236**	92.3973	**96.1235**	50.0603	3.8583
	RF	**96.1235**	**94.2236**	92.3973	**96.1235**	50.0934	3.8439
DeConDFFuse (Ours)	-	90.7422	92.7777	**95.9478**	90.7422	**91.4453**	**34.0847**

However, no single Benchmark has worked well in terms of all the classification metrics used for evaluation. In fact, the next best performance in terms of Accuracy and F1 is given by our method. Despite the highest F1, Accuracy, and recall values, Graph DDI fails to achieve the highest values for AUC -ROC and AUPRC, which are considered more relevant and important metrics for the performance evaluation in the case of binary classification. The reason for the same can be observed with the help of the confusion matrices in Figure 6 that are represented in the form of the percentages.

We can see that with our method, we can predict the highest number of known-to-interact interactions (1) correctly than any other benchmarks. Also, except for Graph DDI, the False positives, i.e., classifying 0 as 1, are lesser with our method than the other two benchmarks. Here, the former task of classifying the known-to-interact drug interactions is more important to prevent ADRs, as explained before and which Graph DDI does not achieve at all or is nearly negligible. Thus, with our method, we are able to accomplish the former task of identifying known-to-interact interactions better than any other benchmark, and for the false positives too, it gives good performance compared to the other benchmarks except graph DDI. The latter is the reason why Graph DDI has the 3 metric values higher than our method. With our method, though the percentage of False positives are higher than Graph DDI, however, it is not necessary that these False positives are completely incorrect. The reason is that the 0 interactions do not signify that there is no interaction between those two drugs. It means either known-not-to-interact or unknown.

In summary, Graph DDI classifies almost all 0 interactions correctly; still, it does not correctly classify 1 (known-to-interact) interactions that are against the study's objective, i.e., to identify the known-to-interact DDIs to avoid ADRs. With our method, both types of interactions are classified reasonably well. It is the stable method corroborated from the classification metrics also as it gives good performance in terms of all the metrics. Hence, our proposed framework performs superior to the benchmarks.

Additionally, let us compare representations / features learned from benchmarks with our framework. Our method performs better due to the kind of representations / features learned from our CTL based network. We carefully examine each of the benchmarks here. In KGNN, Knowledge Graph features, different aggregation techniques and Graph Neural



(a) KGNN - Sum (b) KGNN - Concat (c) KGNN - Aggregate

(d) Conv-LSTM (e) Graph DDI-GNB (f) Graph DDI-LR

(g) Graph DDI-RF (h) Our method - DeConDFFuse

Figure 6: Confusion matrices for different benchmarks and our method- DeConDFFuse

Network are utilized. The latter has a lot of parameters and computation cost since the neural network connects neurons in each layer with every neuron in the preceding and consecutive layers. Also, it has no uniqueness guarantee for the kind of weights learned. The poorest performance from Graph DDI is due to a lack of learning ability of the traditional machine learning algorithms utilized in this framework after learning the embeddings from KG. Lastly, with the Conv-LSTM framework, all the disadvantages of CNN discussed in section 2.4 are applied. Thus, the representations learned with it might have redundancy leading to inferior performance. Therefore, from overall performance, it can be concluded that the representations learned from our method are better than the benchmarks.

The optimizer used for the updating all the parameters of the framework except probability distribution π of decision forest is Adam that uses the automatic differentiation in pytorch for gradient computation. The hyperparameters like learning rate, betas, and eps

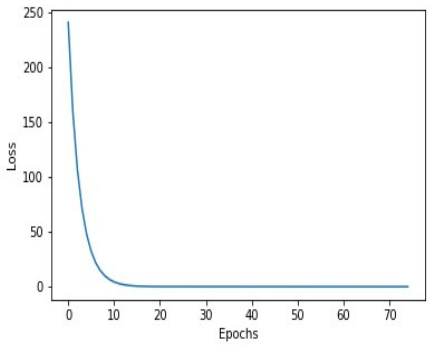


Figure 7: Loss plot with our method - DeConDFFuse.

etc. associated with it are mentioned in the Table 1. We have also plotted the loss plot with our technique, which can be referred to from Figure 7. It can be seen that with Adam optimizer, our solution converges to the point of stability.

6. Conclusion

In this work, we propose applying our recently established work DeConFuse and combining it with Decision Forest previously established in the DNDF framework. The proposed framework is a deep supervised fusion end-to-end framework for processing 1D multi-channel drug data. Unlike other deep learning models that separately use conventional machine learning algorithms like RDF, our framework is jointly optimized and is not piecemeal. We have applied the proposed model for the binary classification task of DDI prediction leading to good performance. The advantage of our framework is its ability to learn unique filters that are not guaranteed with CNNs. It helps us learn non-redundant common representation for the problem where we have two drugs in a drug pair that is not only guided by the deep CTL, but the jointly optimized Decision Forest loss also directs it. We are achieving reasonably well performance compared to the state-of-the-art(s).

The future scope of the work is to improve performance by reducing the number of false positives. Also, the current solution to the DDI problem considers the event when two drugs are administered together. However, combination of more than two drugs are routinely used. Thus, we would also like to extend the capability of our framework to handle more than two drugs' combinations in future. This could be done with our architecture by increasing the number of channels per increase in number of drugs. Lastly, although, we have applied our architecture for drug-drug-interaction, it is flexible enough to be applicable for other biomedical interaction problems. In the future, we will like to explore other areas such as drug-target prediction (Ding et al., 2017; Tanoori et al., 2021; Turki & h. Taguchi, 2019), protein-protein interaction (Sun et al., 2018; Yu et al., 2021; Lee et al., 2006) and drug repositioning (Zhang et al., 2017).

7. Acknowledgement

The authors acknowledge support from Inria Saclay, through the International Inria Team program COMPASS.

References

Abdelaziz, I., Fokoue, A., Zhang, P., & Sadoghi, M. (2017). Large-scale structural and textual similarity-based mining of knowledge graph to predict drug–drug interactions. *Journal of Web Semantics*, *44*, 104–117. URL: https://www.sciencedirect.com/science/article/pii/S157082681730029X. doi:https://doi.org/10.1016/j.websem.2017.06.002.

Allison, M. (2012). Reinventing clinical trials. *Nature biotechnology*, *30(1)*, 41–49. doi:https://doi.org/10.1038/nbt.2083.

Bertoni, M., Duran-Frigola, M., Badia-i Mompel, P., Pauls, E., Orozco-Ruiz, M., Guitart-Pla, O., Alcalde, V., Diaz, V. M., Berenguer-Llergo, A., Brun-Heath, I., Villegas, N., de Herreros, A. G., & Aloy, P. (2021). Bioactivity descriptors for uncharacterized chemical compounds. *Nature Communications*, . doi:https://doi.org/10.1038/s41467-021-24150-4.

Bouvy, J. C., Bruin, M. L. D., & Koopmanschap, M. A. (2015). Epidemiology of adverse drug reactions in europe: a review of recent observational studies. *Drug Safety*, *38(5)*, 437–453.

Celebi, R., Uyar, H., Yasar, E., Gumus, O., Dikenelli, O., & Dumontier, M. (2019). Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings. *BMC Bioinformatics*, *20*, 726. doi:https://doi.org/10.1186/s12859-019-3284-5.

Chen, X., Liu, X., & Wu, J. (2019). Drug-drug interaction prediction with graph representation learning. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 354–361). doi:10.1109/BIBM47256.2019.8983416.

Chen, Y., Ma, T., Yang, X., Wang, J., Song, B., & Zeng, X. (2021). MUFFIN: multi-scale feature fusion for drug–drug interaction prediction. *Bioinformatics*, *37*, 2651–2658. URL: https://doi.org/10.1093/bioinformatics/btab169. doi:10.1093/bioinformatics/btab169.

Dang, L. H., Dung, N. T., Quang, L. X., Hung, L. Q., Le, N. H., Le, N. T. N., Diem, N. T., Nga, N. T. T., Hung, S.-H., & Le, N. Q. K. (2021). Machine learning-based prediction of drug-drug interactions for histamine antagonist using hybrid chemical features. *Cells*, *10*. URL: https://www.mdpi.com/2073-4409/10/11/3092.

Ding, Y., Tang, J., & Guo, F. (2017). Identification of drug-target interactions via multiple information integration. *Information Sciences*, *418-419*, 546–560. URL: https://www.sciencedirect.com/science/article/pii/S0020025517307776. doi:https://doi.org/10.1016/j.ins.2017.08.045.

Duke, J. D., Han, X., Wang, Z., Subhadarshini, A., Karnik, S. D., Li, X., Hall, S. D., Jin, Y., Callaghan, J. T., Overhage, M. J., Flockhart, D. A., Strother, R. M., Quinney, S. K., & Li,

L. (2012). Literature based drug interaction prediction with clinical assessment using electronic medical records: novel myopathy associated drug interactions. *PLoS computational biology*, *8(8)*. doi:https://doi.org/10.1371/journal.pcbi.1002614.

Ferdousi, R., Safdari, R., & Omidi, Y. (2017). Computational prediction of drug-drug interactions based on drugs functional similarities. *Journal of Biomedical Informatics*, *70*, 54–64. URL: https://www.sciencedirect.com/science/article/pii/S1532046417300953. doi:https://doi.org/10.1016/j.jbi.2017.04.021.

Giacomini, K. M., Krauss, R. M., Roden, D. M., Eichelbaum, M., Hayden, M. R., & Nakamura, Y. (2007). When good drugs go bad. *Nature*, *446*, 975–977.

Gupta, P., Maggu, J., Majumdar, A., Chouzenoux, E., & Chierchia, G. (2020). Deconfuse: a deep convolutional transform-based unsupervised fusion framework. *EURASIP J. Adv. Signal Process*, *26*. URL: https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-020-00684-5. doi:https://doi.org/10.1186/s13634-020-00684-5.

Gupta, P., Majumdar, A., Chouzenoux, E., & Chierchia, G. (2021). Superdeconfuse: A supervised deep convolutional transform based fusion framework for financial trading systems. *Expert Systems with Applications*, *169*, 114206. URL: https://www.sciencedirect.com/science/article/pii/S0957417420309349. doi:https://doi.org/10.1016/j.eswa.2020.114206.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* NIPS'17 (p. 1025–1035). Red Hook, NY, USA: Curran Associates Inc.

Huang, D., Jiang, Z., Zou, L., & Li, L. (2017). Drug–drug interaction extraction from biomedical literature using support vector machine and long short term memory networks. *Information Sciences*, *415-416*, 100–109. URL: https://www.sciencedirect.com/science/article/pii/S002002551730110X. doi:https://doi.org/10.1016/j.ins.2017.06.021.

Jonathan H. Watanabe, J. D. H., Terry McInnis (2018). Cost of prescription drug-related morbidity and mortality. *The Annals of pharmacotherapy*, *52(9)*, 829—-837. doi:https://doi.org/10.1177/1060028018765159.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Adv. Neural Inf. Process. Syst.*, *30*, 971—-980.

Kontschieder, P., Fiterau, M., Criminisi, A., & Bulò, S. R. (2015). Deep neural decision forests. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1467–1475). doi:10.1109/ICCV.2015.172.

Lee, H.-C., Huang, S.-W., & Li, E. Y. (2006). Mining protein–protein interaction information on the internet. *Expert Systems with Applications*, *30*, 142–148. URL: https://www.sciencedirect.com/science/article/pii/S0957417405002496. doi:https://doi.org/10.1016/j.eswa.2005.09.083. Intelligent Bioinformatics Systems.

Lin, X., Quan, Z., Wang, Z.-J., Ma, T., & Zeng, X. (2020). Kgnn: Knowledge graph neural network for drug-drug interaction prediction. In *IJCAI*.

Liu, S., Zhang, Y., Cui, Y., Qiu, Y., Deng, Y., Zhang, Z. M., & Zhang, W. (2020). Efficient prediction of drug–drug interaction using deep learning models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *14*, 211–216. doi:https://doi.org/10.1049/iet-syb.2019.0116.

Liu, S., Zhang, Y., Cui, Y., Qiu, Y., Deng, Y., Zhang, Z. M., & Zhang, W. (2022). Enhancing drug-drug interaction prediction using deep attention neural networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (pp. 1–1). doi:https://doi.org/10.1109/TCBB.2022.3172421.

Lounkine, E., Keiser, M., Whitebread, S., Mikhailov, D., Hamon, J., Jenkins, J., Lavan, P., Weber, E., Doak, A., Côté, S., Shoichet, B., & Urban, L. (2012). Large-scale prediction and testing of drug activity on side-effect targets. *Nature*, *486*, 361–367. doi:https://doi.org/10.1038/nature11159.

Mongia, A., Jain, S., Chouzenoux, É., & Majumdar, A. (2020). Deepvir - graphical deep matrix factorization for "in silico" antiviral repositioning: Application to covid-19. *ArXiv*, *abs/2009.10333*.

Park, C., Park, J., & Park, S. (2020). Agcn: Attention-based graph convolutional networks for drug-drug interaction extraction. *Expert Systems with Applications*, *159*, 113538. URL: https://www.sciencedirect.com/science/article/pii/S0957417420303626. doi:https://doi.org/10.1016/j.eswa.2020.113538.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch. *NIPS Autodiff Workshop*, .

Ravishankar, S., & Bresler, Y. (2013). Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, *61*, 1072–1086. doi:https://www.doi.org/10.1109/TSP.2012.2226449.

Rezaul Karim, M., Cochez, M., Jares, J., Uddin, M., Beyan, O., & Decker, S. (2019). Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network. In *ACM-BCB 2019 - Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* ACM-BCB 2019 - Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (pp. 113–123). Association for Computing Machinery, Inc. doi:https://doi.org/10.1145/3307339.3342161.

Rohani, N., & Eslahchi, C. (2019). Drug-drug interaction predicting by neural network using integrated similarity. *Sci Rep*, *9*. doi:https://doi.org/10.1038/s41598-019-50121-3.

Sahu, S. K., & Anand, A. (2018). Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, *86*, 15–24. URL: https://www.sciencedirect.com/science/article/pii/S1532046418301606. doi:https://doi.org/10.1016/j.jbi.2018.08.005.

Sandritter, T. L., Jones, B. L., Kearns, G. L., & Lowry, J. A. (). *Nelson Textbook of Pediatrics*.

Shi, J.-Y., Huang, H., Li, J.-X., Lei, P., Zhang, Y.-N., Dong, K., & Yiu, S.-M. (2018). Tmfuf: a triple matrix factorization-based unified framework for predicting comprehensive drug-drug interactions of new drugs. *BMC Bioinformatics*, *19*, 411. doi:https://doi.org/10.1186/s12859-018-2379-8.

Shi, J.-Y., Huang, H., Li, J.-X., Lei, P., Zhang, Y.-N., & Yiu, S.-M. (2017). Predicting comprehensive drug-drug interactions for new drugs via triple matrix factorization. In I. Rojas, & F. Ortuño (Eds.), *Bioinformatics and Biomedical Engineering* (pp. 108–117). Cham: Springer International Publishing.

Sridhar, D., Fakhraei, S., & Getoor, L. (2016). A probabilistic approach for collective similarity-based drug–drug interaction prediction. *Bioinformatics*, *32*, 3175–3182. doi:https://doi.org/10.1093/bioinformatics/btw342.

Sun, P. G., Quan, Y. N., Miao, Q. G., & Chi, J. (2018). Identifying influential genes in protein–protein interaction networks. *Information Sciences*, *454-455*, 229–241. URL: https://www.sciencedirect.com/science/article/pii/S0020025518303487. doi:https://doi.org/10.1016/j.ins.2018.04.078.

Tanoori, B., Jahromi, M. Z., & Mansoori, E. G. (2021). Drug-target continuous binding affinity prediction using multiple sources of information. *Expert Systems with Applications*, *186*, 115810. URL: https://www.sciencedirect.com/science/article/pii/S0957417421011787. doi:https://doi.org/10.1016/j.eswa.2021.115810.

Turki, T., & h. Taguchi, Y. (2019). Machine learning algorithms for predicting drugs–tissues relationships. *Expert Systems with Applications*, *127*, 167–186. URL: https://www.sciencedirect.com/science/article/pii/S0957417419301186. doi:https://doi.org/10.1016/j.eswa.2019.02.013.

Whitebread, S., Hamon, J., Bojanic, D., & Urban, L. (2005). Keynote review: In vitro safety pharmacology profiling: an essential tool for successful drug development. *Drug Discov Today*, *10(21)*, 1421—-1433. doi:https://doi.org/10.1016/S1359-6446(05)03632-9.

Wu, H., Xing, Y., Ge, W., Liu, X., Zou, J., Zhou, C., & Liao, J. (2020). Drug-drug interaction extraction via hybrid neural networks on biomedical literature. *Journal of Biomedical Informatics*, *106*, 103432. URL: https://www.sciencedirect.com/science/article/pii/S1532046420300605. doi:https://doi.org/10.1016/j.jbi.2020.103432.

Yu, B., Chen, C., Wang, X., Yu, Z., Ma, A., & Liu, B. (2021). Prediction of protein–protein interactions based on elastic net and deep forest. *Expert Systems with Applications*, *176*, 114876. URL: https://www.sciencedirect.com/science/article/pii/S0957417421003171. doi:https://doi.org/10.1016/j.eswa.2021.114876.

Yu, H., Dong, W., & Shi, J. (2022). Raneddi: Relation-aware network embedding for drug-drug interaction prediction. *Information Sciences*, *582*, 167–180. URL: https://www.sciencedirect.com/science/article/pii/S0020025521009294. doi:https://doi.org/10.1016/j.ins.2021.09.008.

Yu, M., Kim, S., Wang, Z., Hall, S., & Li, L. (2008). A bayesian meta-analysis on published sample mean and variance pharmacokinetic data with application to drug–drug interaction prediction. *Journal of Biopharmaceutical Statistics*, *18*, 1063–1083. doi:https://doi.org/10.1080/10543400802369004. arXiv:https://doi.org/10.1080/10543400802369004.

Zhang, F., Sun, B., Diao, X., Zhao, W., & Shu, T. (2021). Prediction of adverse drug reactions based on knowledge graph embedding. *BMC Medical Informatics and Decision Making*, *21*. doi:https://doi.org/10.1186/s12911-021-01402-3.

Zhang, J., Li, C., Lin, Y., Shao, Y., & Li, S. (2017). Computational drug repositioning using collaborative filtering via multi-source fusion. *Expert Systems with Applications*, *84*, 281–289. URL: https://www.sciencedirect.com/science/article/pii/S0957417417303202. doi:https://doi.org/10.1016/j.eswa.2017.05.004.

Zhang, W., Chen, Y., Li, D., & Yue, X. (2018). Manifold regularized matrix factorization for drug-drug interaction prediction. *Journal of Biomedical Informatics*, *88*, 90–97. URL: https://www.sciencedirect.com/science/article/pii/S1532046418302144. doi:https://doi.org/10.1016/j.jbi.2018.11.005.

Zhang, W., Jing, K., Huang, F., Chen, Y., Li, B., Li, J., & Gong, J. (2019). Sflln: A sparse feature learning ensemble method with linear neighborhood regularization for predicting drug–drug interactions. *Information Sciences*, *497*, 189–201. URL: https://www.sciencedirect.com/science/article/pii/S0020025519304116. doi:https://doi.org/10.1016/j.ins.2019.05.017.

Zhang, Y., Qiu, Y., Cui, Y., Liu, S., & Zhang, W. (2020). Predicting drug-drug interactions using multi-modal deep auto-encoders based network embedding and positive-unlabeled learning. *Methods*, *179*, 37–46. URL: https://www.sciencedirect.com/science/article/pii/S1046202319303421. doi:https://doi.org/10.1016/j.ymeth.2020.05.007. Interpretable machine learning in bioinformatics.