

Detecting drug-drug interactions using artificial neural networks and classic graph similarity measures

Guy Shtar,^{1*} Lior Rokach,¹ Bracha Shapira,¹

¹ Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel

* Shtar@post.bgu.ac.il

Abstract

Drug-drug interactions are preventable causes of medical injuries and often result in doctor and emergency room visits. Computational techniques can be used to predict potential drug-drug interactions. We approach the drug-drug interaction prediction problem as a link prediction problem and present two novel methods for drug-drug interaction prediction based on artificial neural networks and factor propagation over graph nodes: adjacency matrix factorization (AMF) and adjacency matrix factorization with propagation (AMFP). We conduct a retrospective analysis by training our models on a previous release of the DrugBank database with 1,141 drugs and 45,296 drug-drug interactions and evaluate the results on a later version of DrugBank with 1,440 drugs and 248,146 drug-drug interactions. Additionally, we perform a holdout analysis using DrugBank. We report an area under the receiver operating characteristic curve score of 0.807 and 0.990 for the retrospective and holdout analyses respectively. Finally, we create an ensemble-based classifier using AMF, AMFP, and existing link prediction methods and obtain an area under the receiver operating characteristic curve of 0.814 and 0.991 for the retrospective and the holdout analyses. We demonstrate that AMF and AMFP provide state of the art results compared to existing methods and that the ensemble-based classifier improves the performance by combining various predictors. These results suggest that AMF, AMFP, and the proposed ensemble-based classifier can provide important information during drug development and regarding drug prescription given only partial or noisy data. These methods can also be used to solve other link prediction problems. Drug embeddings (compressed representations) created when training our models using the interaction network have been made public.

Introduction

Adverse drug events are often preventable causes of medical injuries, and adverse drug reactions (ADRs) are estimated to be the fourth leading cause of death in the U.S., ahead of pulmonary disease, diabetes, AIDS, pneumonia, accidents, and automobile fatalities [1]. The cost attributed to ADRs is estimated to be over \$1,000 per patient per year in the US [2]. Estimates of the number of patients harmed due to drug interactions range from 3-5% of all medication errors within hospitals. Additionally, drug interactions are the cause of many patient visits to physicians and emergency units [3,4]. Thirty-six percent of older adults in the U.S. regularly use five or more medications or supplements, and 15% are potentially at risk for a major drug-drug

interaction (DDI) [5]. The American Geriatrics Society has identified the consideration of drug-disease and drug-drug interactions as a key element of optimal care for older adults with multimorbidity [6]. DDI prediction during the clinical experiments conducted in order to approve a new drug is difficult [7]. Clinical trials for new drugs don't address the issue of DDI directly, and potential DDIs are often not discovered until the third phase of a clinical trial or once the drug is already on the market. The most practical way to explore the large number of drug combinations for detecting interacting drugs is through in silico drug-drug interaction detection, and in this paper, we propose a computational method for DDI detection.

In recent years, the detection of potential DDIs using computational techniques has gained attention; previous research has used techniques based on drug-drug interaction similarities [8], side effect similarities [9], structural similarities [10], or a combination of various similarity measures [11–13]. Other works use natural language processing (NLP) techniques to train word embedding using document collections such as PubMed, PMC, MEDLINE, and Wikipedia; the embeddings are later used to predict DDIs [14]. Computational methods often require a large amount of data for optimization. For example, when evaluating a new drug using structural-based similarity methods, the method will require data showing a strong, well established history for structurally similar drugs in order to accurately detect drug interactions. Side effect similarity-based methods require data for drugs with similar side effects, etc. Therefore, the drug-drug interaction prediction problem should be investigated and tackled using various data types.

DDI detection can be seen as a special case of link prediction in a graph. In a link prediction problem, we seek to accurately predict the edges (interactions) between nodes (drugs) that will be added to the network. We approach the DDI prediction problem as a link prediction problem. Perhaps the most basic approach is to rank edges based on the idea that two nodes x and y are more likely to form a link if their sets of neighbors have a large overlap; this follows the natural intuition that such nodes x and y represent drugs with many interacting drugs in common, and hence are more likely to interact. Matrix factorization is another approach for resolving link prediction problems. Matrix factorization (MF) is the factorization of a matrix into a product of matrices; this technique is widely used for dimensionality reduction, specifically in the field of recommender systems. In recent years, successful attempts have been made to factorize a matrix using deep neural networks [15–17]. In this paper, we introduce AMF and AMFP, two novel methods for predicting DDIs based on an artificial neural networks and the implementation of factor propagation over the interaction network. AMF and AMFP take known drug interactions as input and predict currently unknown drug interactions. We compare AMF and AMFP to existing methods and create an ensemble-based classifier using AMF, AMFP, and other well-known link prediction methods. In this paper, we make three key contributions: (1) we formulate a new artificial neural network-based method for link prediction, (2) we demonstrate its effectiveness for the drug-drug interaction prediction challenge, conducting extensive evaluations with real data to show the superiority of our method, and create drug embeddings for all available drugs, and (3) we create an ensemble-based classifier to demonstrate the benefit of combining existing high-performing classifiers. The preprocessing, methods, and drug embeddings developed, calculated, and used in this research were implemented and have been made public: https://github.com/goolig/DDI_prediction.

Materials and methods

Problem formulation

We approach the drug-drug interaction prediction problem as a link prediction problem. Suppose we have an undirected drug interaction network $G = (V, E)$ in which each edge $e = (u, i) \in E$ represents an interaction between drugs u and i . Note that throughout the paper we use the terms graph node and drug interchangeably. We use two versions of the drug interactions graph: G and G' . For two time snapshots $t < t'$ let G denote the graph constructed using the known interactions at time t , and G' denote the constructed graph using the known interactions at time t' . This is a concrete formulation of the drug-drug interaction prediction problem: we give an algorithm access to network G , and it must then output a list of edges not present in G , which are predicted to appear in network G' . We refer to t as the training release date and t' as the test release date. Of course, drug interaction networks grow through the addition of nodes (drugs) as well as edges. The training process uses only existing interactions to predict unknown ones. It is not sensible to seek predictions for edges whose endpoints are not present in the training interval, as such a prediction will be based on partial information in terms of link prediction. We use the adjacency matrix to represent G and G' , and the matrices are used to train AMF and AMFP: Let M denote the number of drugs. We define the drug-drug interaction matrix $Y \in \mathbb{R}^{M \times M}$ as follows:

$$y_{i,j} = \begin{cases} 1, & \text{if interaction exists between drugs } i,j; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Here, a value of one for $y_{i,j}$ indicates an existing interaction between drugs i and j , however a value of zero does not mean that an interaction does not exist - it could be that the interaction has not yet been discovered.

Adjacency matrix factorization

Traditionally, matrix factorization associates each row element i and column element j with a corresponding latent vector p_i and q_j . The estimate of the corresponding cell $y_{i,j}$ of the matrix is given by the inner product of the vectors:

$$\hat{y}_{i,j} = \sum_{w=1}^k p_{iw} q_{jw}, \quad (2)$$

where the vectors p_i and q_j are the latent factors, sometimes also referred to as representations, because they can be used as an alternative representation of the original row and column objects. The space size k is a parameter, usually set to a much lower value than the original space size. Using an extremely low k value might lead to underfitting; on the other hand, extremely high values might lead to overfitting. MF is sometimes improved by using a bias value corresponding to the row and column elements:

$$\hat{y}_{i,j} = \mu + b_i + b_j + \sum_{w=1}^k p_{iw} q_{jw}, \quad (3)$$

where μ is the average value over the whole matrix, and b_i and b_j are the bias values for the row and column elements, correspondingly. The parameters are typically learned using optimization techniques such as stochastic gradient descent. Regularization techniques are often used during the optimization process. AMF (adjacency matrix factorization) performs matrix factorization on the adjacency matrix

of G . Because the graph G is undirected, the adjacency matrix M is symmetric. Therefore, it is sufficient to use a single vector and bias value shared between the rows and columns to estimate M 's cells. To allow precise DDI prediction, we use an artificial neural network-based model that encompasses the linear structure of the interaction network. The method is based on optimizing the latent factors of each drug in the network; the latent factor is an k -dimensional vector. Figure 1 provides an overview of AMF's architecture. AMF takes a one-hot encoding representation of the two nodes under consideration as input. The output of AMF is binary; one indicates an existing interaction, and zero indicates no interaction between the two input nodes. Note that only an existing drug interaction network is required to train the proposed neural network. No other domain-specific information is required.

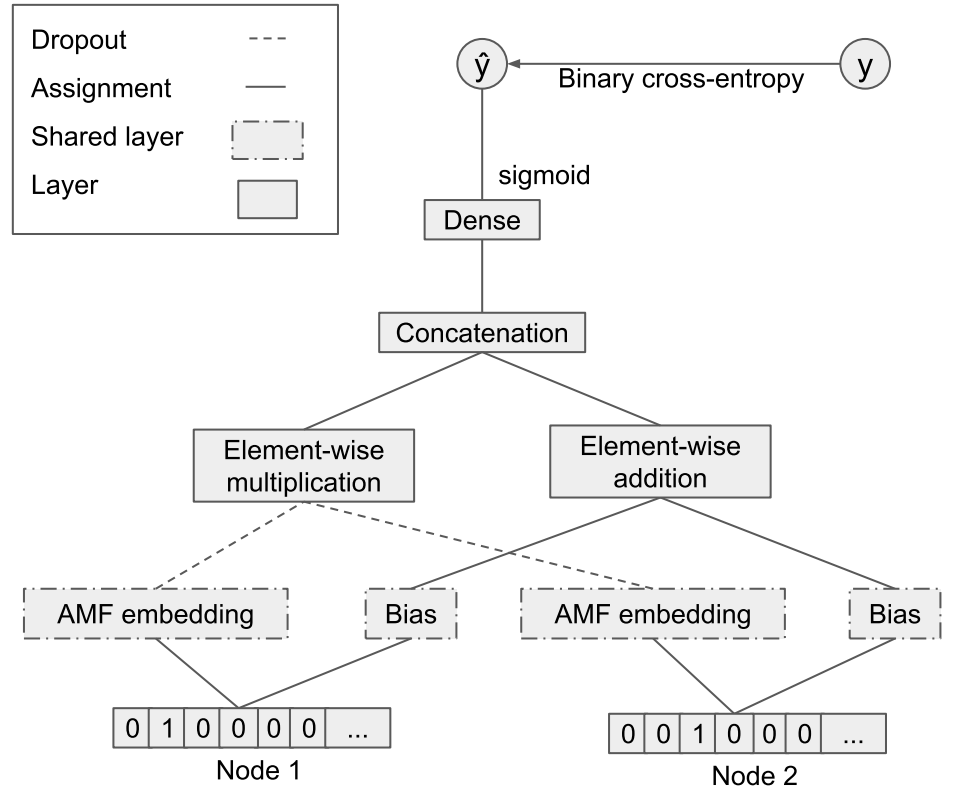


Fig 1. Overview of AMF's architecture. Drugs are represented as nodes; embedding layers (which act as latent factors) and biases are shared between input nodes. Dropout is used as a regularization mechanism for preventing overfitting.

The use of a simple inner product to estimate complex drug-drug interactions in the low-dimensional latent space might be oversimplistic. The complexity of this model can be increased by increasing the embedding layer size, however that might cause overfitting. The space size k should be carefully tuned during the training phase. Matrix factorization and AMF are closely related to singular value decomposition (SVD). AMF's main improvements in the optimization stage compared to matrix factorization and SVD are: (1) sharing the latent vectors of the rows and columns, and (2) optimizing the weights of the element-wise multiplication and the bias, rather than just using a dot product and adding the biases. AMF is optimized using the

adaptive moment estimation (Adam) optimization algorithm [18] which adjusts the learning rate for each parameter using estimates of the first and second moments of the gradients. MF and SVD are aimed at minimizing the mean square error; for a classification task such as a link prediction problem, binary cross-entropy is more appropriate. Hence, the loss function used in AMF is binary cross-entropy, defined as follows:

$$L = \sum_{i,j \in Y} -y_{i,j} \log \hat{y}_{i,j} - (1 - y_{i,j}) \log (1 - \hat{y}_{i,j}), \quad (4)$$

where Y is the set of instances (drug pairs), $y_{i,j}$ is the true label which represents the existence or absence of an interaction, and $\hat{y}_{i,j}$ is the predicted value. Y is created using negative sampling, where all positive samples are used (existing drug interactions), and a relative number of negative samples are drawn randomly in each epoch; the number of negative samples to be sampled is a hyperparameter of the model that should be tuned. In this research we sample one negative sample for each given positive sample.

Adjacency matrix factorization with propagation

AMF excels in the holdout analysis where data is randomly sampled for the testing set (see Results section for more details). Nevertheless, despite all of the regularization techniques applied to it, AMF’s generalization ability and performance are poor in the retrospective analysis where a new unseen version of the dataset is used. The underlying mechanism that causes an interaction to be discovered and added to the database is not random - it depends on mediators such as new interactions discovered between substances contained in the drugs, the drug’s prevalence, and more. To perform well in the retrospective analysis, higher generalization ability is required from the model. Adjacency matrix factorization with propagation (AMFP) is an extension of AMF. In AMFP, the same model used in AMF is used, but an additional step is performed: propagating the latent factors of each drug to its interacting drugs; latent factor propagation is controlled by a propagation factor, which controls the weight of the original latent factor (which was optimized in the previous step) compared to the weight of the latent factor of the interacting drugs. Algorithm 1 describes the propagation procedure. Each node’s latent factor is shared with the node’s neighborhood. The parameter α is the propagation factor, which controls how much information will be passed from the neighboring nodes. The value of α should be optimized during the training process.

Algorithm 1 Latent factor propagation

```

1: procedure PROPAGATE_FACTORS(Graph  $G = (V, E)$ , Latent factors  $P$ , Propagation
   factor  $\alpha$ )
2:    $P' \leftarrow \text{empty list}$  // holds the new latent factors
3:   for vertex  $v1$  in  $G$  do
4:      $Q \leftarrow \text{empty latent factor}$ 
5:     for vertex  $v2$  in  $\Gamma(v1)$  do
6:        $Q \leftarrow Q + \frac{1}{|\Gamma(v1)|} \cdot P_{v2}$ 
7:     end for
8:      $P'_{v1} \leftarrow \alpha \cdot Q + (1 - \alpha) \cdot P_{v1}$ 
9:   end for
10:  return  $P'$ 
11: end procedure

```

Given a node $v \in V$; the neighborhood of v is defined by $\Gamma(v)$ and represented by the set of v 's interacting drugs. The lists P and P' contain the original latent factors (embeddings) and the latent factors resulting from the propagation process correspondingly. Each list contains vectors (each vector has k elements) representing the latent factor of the nodes. α is the propagation factor; when α reaches a value of one, the original latent factor of each node is discarded, and a new latent factor is created based on the neighborhood of the node. On the other extreme, when α reaches a value of zero, the latent factors created in the previous step are used, and the propagation step does not change the model. When α 's value is equal to zero, the results of AMPF and AMF are equivalent. Propagating the factors is expected to improve the generalization ability of the model by combining the factors of interacting drugs. This logic follows the assumption that interacting drugs share some common characteristics.

Link prediction similarity measures

Link prediction similarity measures can be viewed as pre-engineered features which leverage domain knowledge for link prediction in graphs. This subsection is devoted to formulating and explaining the motivation behind the similarity measures used in this research for creating an ensemble-based classifier and evaluation.

The common neighbors between two given nodes $u, v \in V$ refers to the size of the set of common neighbors that both u and v possess. The formal common neighbors definition is:

$$S^{CN}(u, v) = |\Gamma(v) \cap \Gamma(u)|. \quad (5)$$

The relevance of the common neighbors feature is very intuitive. It is expected that the larger the size of the common neighborhood, the higher the chances are that both vertices will be connected. The common neighbors feature has been widely used in past work on link prediction on several datasets and was found to be very helpful [19]. Using the common neighbors measure, we formulate the average common neighbors for two nodes:

$$\tilde{S}^{ACN}(u, v) = \frac{1}{|\Gamma(v)|} \sum_{w \in \Gamma(v)} S^{CN}(w, u). \quad (6)$$

The average common neighbors measure provided above is not symmetric, due to the normalizing factor $|\Gamma(v)|$; we formulate it as a symmetric measure by averaging its two possible values for a pair of nodes:

$$S^{ACN}(u, v) = \frac{\tilde{S}^{ACN}(u, v) + \tilde{S}^{ACN}(v, u)}{2}. \quad (7)$$

The Jaccard coefficient is a well-known similarity measure, widely used for link prediction [19]. For two nodes $u, v \in V$ the Jaccard coefficient is defined as follows:

$$S^{Jaccard}(u, v) = \frac{|\Gamma(v) \cap \Gamma(u)|}{|\Gamma(v) \cup \Gamma(u)|}. \quad (8)$$

As with common neighbors, we formulate the average Jaccard coefficient for two vertices as follows:

$$\tilde{S}^{AJ}(u, v) = \frac{1}{|\Gamma(v)|} \sum_{w \in \Gamma(v)} S^{Jaccard}(w, u), \quad (9)$$

and its symmetric version is given as follows:

$$S^{AJ}(u, v) = \frac{\tilde{S}^{AJ}(u, v) + \tilde{S}^{AJ}(v, u)}{2}. \quad (10)$$

The Adamic/Adar index [20] is a similarity measure used to predict links in social networks:

$$S^{AA}(u, v) = \sum_{w \in \Gamma(v) \cap \Gamma(u)} \frac{1}{\log |\Gamma(w)|}. \quad (11)$$

Lastly, we present $Katz_b$ which exponentially sums the number of shortest paths of different lengths between two nodes:

$$Katz_b(u, v) = \sum_{\ell=1}^{\infty} \beta^{\ell} \cdot |paths|_{u,v}^{<\ell>}, \quad (12)$$

where $|paths|_{u,v}^{<\ell>}$ is the number of paths between u and v of length ℓ , and β is a parameter controlling the weight given to shorter paths compared to the weight given to longer ones. In practice, a truncated Katz measure is usually used:

$$Katz_b(u, v) = \sum_{\ell=1}^b \beta^{\ell} \cdot |paths|_{u,v}^{<\ell>}. \quad (13)$$

In this research, we use $b = 3$ due to resource limitations. $Katz_b$ was found to be very helpful for link prediction in previous works [21]. The final list of link prediction similarity measures used in this research consists of: average common neighbors, average Jaccard coefficient, Adamic/Adar, and $Katz_b$. The similarity measures between two nodes used by Fire et al. [22], such as the shortest path length between nodes, cosine distance between nodes, and dividing the graph into communities and then comparing two nodes' communities were tested and discarded due to poor performance.

Creating an ensemble-based classifier

Ensembles are meta-algorithms used to combine various classifiers. They can reduce the variance and bias of the base models and improve the predictions in general. One such ensemble method is XGBoost [23] which achieved state of the art results in multiple tasks and competitions. XGBoost employs gradient boosting where models are created stage wise using weak predictors, usually using prediction trees. In each stage, the model seeks to improve the performance of the model created in the previous stage. We train an ensemble classifier using XGBoost, based on the link prediction similarity measures presented above: average common neighbors, average Jaccard coefficient, Adamic/Adar, and $Katz_b$. Additionally AMF or AMFP (the method that performs better) and the method proposed by Vilar et al. [8] are fed to the ensemble classifier. This meta-algorithm can be easily extended to include additional features.

Evaluation

In this section, we present experiments with the aim of evaluating AMF, AMFP, and the ensemble-based classifier. Our evaluation is based on two evaluation schemes: a retrospective analysis using approved drugs from three versions of the DrugBank database [24] and a holdout analysis using a current version of the database. We use state of the art benchmarks.

Figure 2 illustrates the validation and testing scheme for the retrospective analysis using three versions of the DrugBank database. Major changes were made between the versions - specifically, a large number of interactions were added to the more recent version. For the validation process, we aligned versions 4.1.0 and 5.0.0 by using only drugs which appear in both versions. The same was done for versions 5.0.0 and 5.1.1 when training and testing the final model. Version 4.1.0 from December 2014 contains 11,284 interactions, and versions 5.0.0 from June 2016 and 5.1.1 from July 2018

contain 45,296 and 248,146 interactions respectively. Versions 4.1.0, 5.0.0, and 5.1.1 respectively contain 1,141, 1,440, and 2,149 drugs.

In addition to the retrospective analysis, we perform a holdout evaluation using release 5.1.1, the latest release available during this research. The setup of the holdout evaluation is as follows: 30% of randomly selected existing and non-existing interactions are used as a test set, and the rest of the data is used as a training set, 10% of the data is used for validation (parameter tuning) during training. For both evaluation techniques, the model is retrained after the validation process, with the combined training and validation data, using the tuned parameters. In a holdout evaluation, the interactions are randomly selected, while in reality some interactions are more likely to be found earlier depending on the popularity of the drug, the prevalence of the interaction, etc. For these reasons, the retrospective analysis is a stronger evaluation scheme, however we perform a holdout evaluation for comprehensiveness and to comply with previous research.

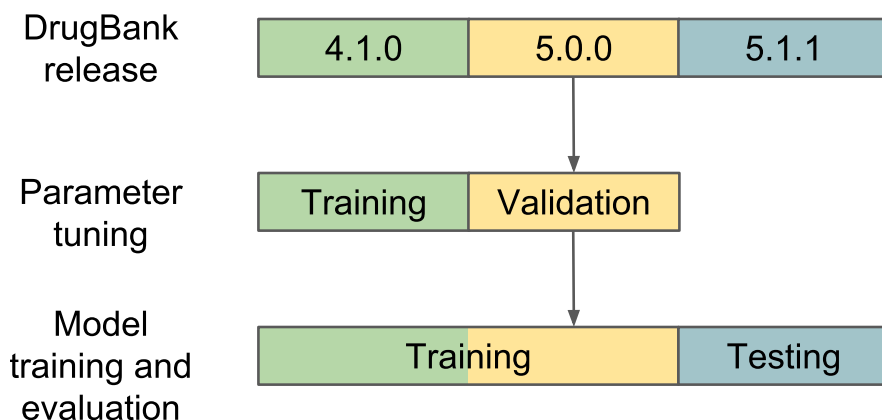


Fig 2. Retrospective evaluation scheme. Parameter tuning is performed using DrugBank release 4.1.0 and 5.0.0. The previous release is used to train the model, and the latter is used to validate the results. The final model is trained using the parameters obtained in the validation stage with the data from release 5.0.0 (which contains the data from release 4.1.0 with some additions and changes) and tested using release 5.1.1.

Metrics

The primary evaluation metric we use is the area under the ROC curve (AUROC). We also assess the area under the precision-recall curve (AUPR), because it was argued to be relevant for link prediction problems [25]. We plot the ROC curve and the average precision @ n , where the precision of each drug’s prediction is averaged at different values of n . Lastly, we plot precision @ n which evaluates the top n most confident predictions of the model. We acknowledge the importance of precision over recall in the DDI problem, therefore we plot the two precision graphs in addition to the other metrics.

Baselines

We compare our proposed method with the following methods:

- The method suggested by Vilar et al. [8]. This method is based on drug interaction profile fingerprints (IPFs). The model uses IPFs to measure the similarity of pairs of drugs and generates new putative drug-drug interactions from the non-intersecting interactions of a pair. Their method uses the same input data used by the method proposed in this paper.
- The link prediction similarity measures presented earlier: average common neighbors, average Jaccard coefficient, Adamic/Adar, and $Katz_b$.
- An XGBoost model trained using all of the models described in the previous bullets and AMF or AMFP (we use the model which performs better). This model is used to demonstrate the power of combining several strong methods, rather than being used in comparison with the other methods mentioned above, as a comparison between a regular model and an ensemble is inappropriate.

We implemented AMF and AMFP using Keras [26]. The method suggested by Vilar et al. was implemented in Python; for Adamic/Adar and several other methods we used the NetworkX implementation [27]. Other methods, such as $Katz_b$, were implemented in Python. For XGBoost, the implementation proposed by the authors was used. We assess the AUROC score of AMF, AMFP, and each of the baselines for significance with a paired test using the algorithm described by Sun and Xu [28].

Parameter tuning

To determine the hyperparameters of AMF and AMFP, we used the procedure described above. All weights are randomly initialized using the Glorot normal initializer [29]. The following batch sizes were used: 128, 256, 512, and 1024, and learning rates in the range of 0.1 - 0.0001 were tested. We evaluate the following number of factors (embedding sizes): 32, 64, 128, 256, 512, and 1024, dropout levels in the range of 0-0.9, the number of epochs in the range of 1-10, and propagation factors in the range of 0.0-1.0. For XGBoost, the parameters were optimized using randomized grid search, where combinations of parameters were drawn randomly from a given list and evaluated.

Results

In this section, we report the results of AMF, AMFP, and the baselines using the evaluation techniques described in the previous section.

Holdout analysis

Holdout analysis is performed by using 70% of the data in DrugBank release 5.1.1 to train the models; the rest of the existing and non-existing interactions are used for evaluation. Fig 3A shows the ROC curve for AMF and each of the baselines. Table 1 presents the AUROC and AUPR values for each model. The AUROC of each pair of models was tested for significance; we report a $p - value < 10^{-4}$ for all tests. Figure 3B shows the average precision @ n (per drug), where n ranges from one to five, and figure 3C shows the precision @ n , where n ranges from one to 100. The optimal value for AMFP's α is zero, hence its performance is equivalent to that of AMF. Therefore, we do not present its results in the holdout analysis, and it is not used in the XGBoost model trained using the holdout data.

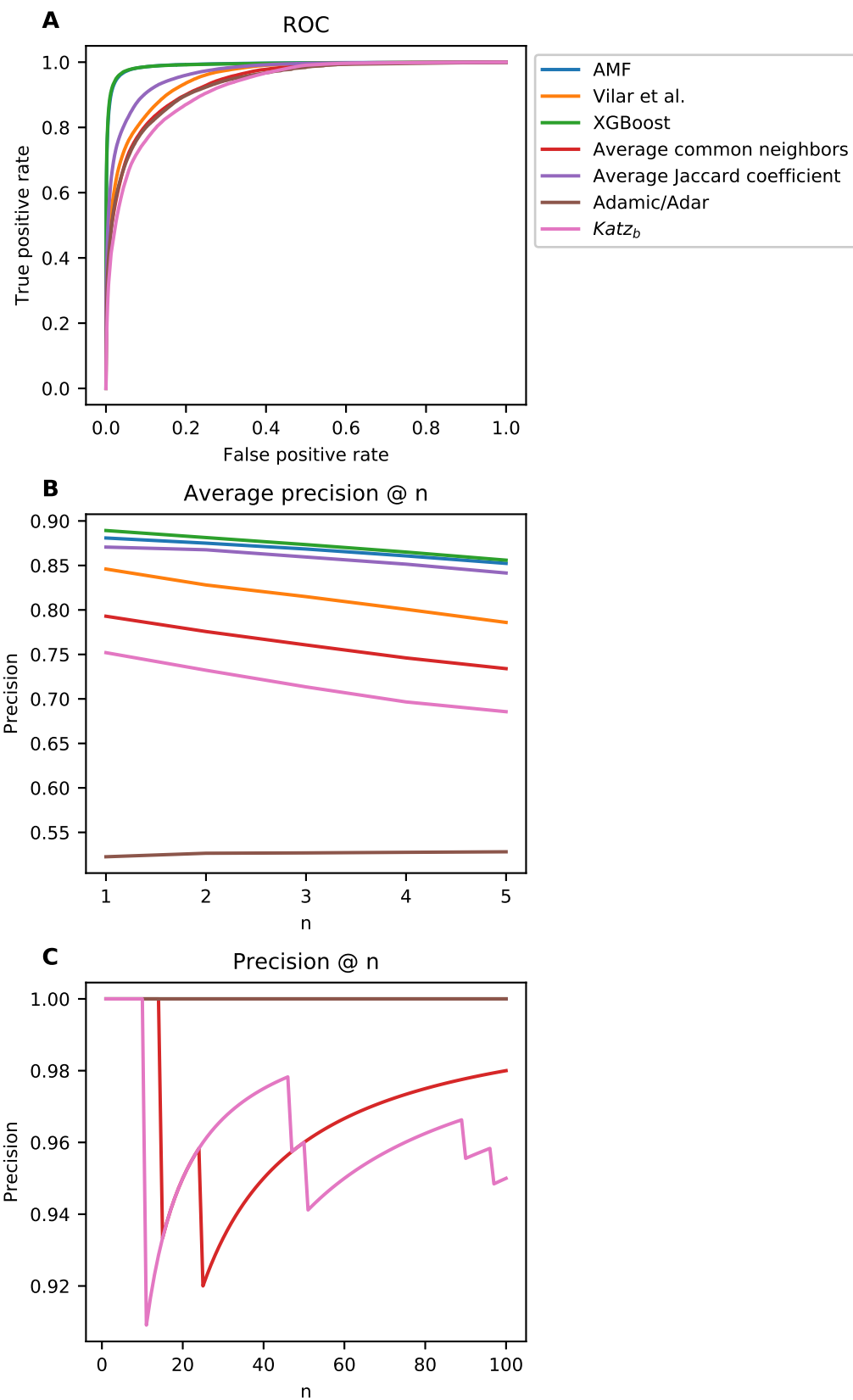


Fig 3. Holdout analysis results. **A)** Receiver operating characteristic curves; **B)** Per drug average precision @ n; **C)** Precision @ n.

Table 1. Area under the ROC and precision-recall curves for the holdout analysis.

Algorithm	AUROC curve	AUPR curve
<i>XGBoost</i>	0.991	0.960
<i>AMF</i>	0.990	0.950
<i>Vilar et al. [8]</i>	0.952	0.784
<i>Average common neighbors</i>	0.938	0.738
<i>Average Jaccard coefficient</i>	0.967	0.840
<i>Adamic/Adar</i>	0.933	0.728
<i>Katz_b</i>	0.924	0.675

Retrospective analysis

Retrospective analysis is performed by training the models on an older version of DrugBank and evaluating the models using a more recent version of DrugBank. Fig 4A shows the ROC curve for AMF, AMFP, and each of the baselines. Table 2 presents the AUROC and AUPR values for each model. The AUROC of each pair of models was tested for significance; we report a $p - value < 10^{-4}$ for all tests. Figure 4B shows the average precision @ n (per drug), where n ranges from one to five, for the first interaction ($n = 1$), the accuracy is about 56% for both AMFP and the XGBoost ensemble. Figure 4C shows the precision @ n , where n ranges from one to 100. XGBoost was trained using AMFP’s predictions and without AMF’s predictions because of their superiority. XGBoost has the best performance in terms of the AUROC and AUPR curves, followed by AMFP. The average precision @ n and precision @ n graphs demonstrate the XGBoost model superiority, it performs best for almost all values of n . While the relative performance between the retrospective analysis and the holdout analysis is somewhat similar for all models, the absolute differences are obvious. This phenomena can be explained by the fact that each DrugBank release is a closed system of interactions known at a given time, sometimes derived from interactions between substances contained in different drugs. The absolute differences between results demonstrate the weakness of holdout evaluation and its difficulty in simulating real-world scenarios compared to retrospective evaluation.

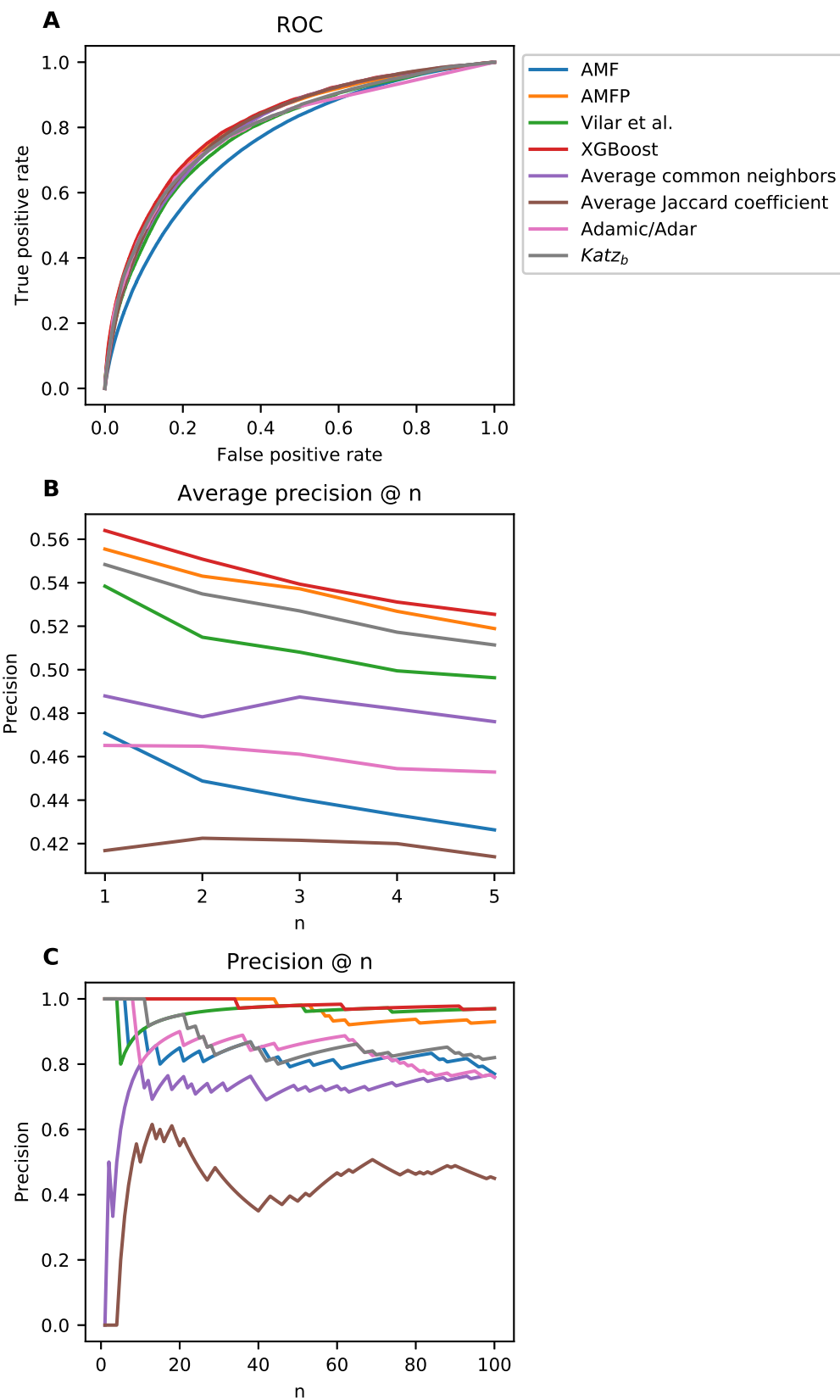


Fig 4. Retrospective analysis results. A) Receiver operating characteristic curves; B) Per-drug average precision @ n; C) Precision @ n.

Table 2. Area under the ROC and precision-recall curves for retrospective analysis.

Algorithm	AUROC curve	AUPR curve
<i>XGBoost</i>	0.814	0.425
<i>AMFP</i>	0.807	0.417
<i>AMF</i>	0.748	0.304
<i>Vilar et al. [8]</i>	0.787	0.38
<i>Average common neighbors</i>	0.802	0.385
<i>Average Jaccard coefficient</i>	0.804	0.370
<i>Adamic/Adar</i>	0.791	0.388
<i>Katz_b</i>	0.798	0.395

Discussion

Drug interactions are the cause of many patient visits to physicians and emergency units. Estimates of the number of patients harmed due to drug interactions range from 3-5% of all medication errors within hospitals [3, 4]. Potential DDIs are often not discovered until the third phase of a clinical trial or in many cases, only after the drug has already been on the market for some time. In silico drug-drug interaction prediction methods, such as the methods proposed in the current research, are the most practical way of detecting DDIs. We introduced AMF and AMFP, two new methods for in silico drug-drug interaction prediction and used DrugBank to demonstrate the superiority of the proposed methods compared to existing methods for the following metrics: AUROC and AUPR curves, precision @ n, and average precision @ n per drug. The improvement was demonstrated by predicting the interactions for a new version of DrugBank and when using a holdout evaluation scheme. An ensemble method trained using XGBoost obtained better results than AMF and AMFP in most metrics (AUROC, AUPR and average precision @ n per drug), and its results are best for almost all values of n in the precision @ n graph. Potentially, the XGBoost ensemble can be further improved by adding more models or including domain specific information (e.g., structural data), however this may come at the cost of a much longer training time.

In this section, we present a more in depth analysis on the propagation factor of AMFP, testing different values for the validation and test sets to investigate whether there is a strong correlation between the two. Figure 5 presents AMFP’s propagation factor analysis. A big difference can be seen between the two evaluation schemes. For the retrospective scheme the optimal values are 0.5 and 0.8 respectively for the validation and test sets. The effect of different propagation factor values on the AUROC of the validation and test sets is similar. For the holdout evaluation scheme the effect of different propagation factor values on the AUROC of the validation and test sets is very similar; smaller values are preferred, and zero is the optimal value. This means that no propagation is required for the validation or test sets. This difference could be the result of the difference in the test set distributions. As stated in the Evaluation section, retrospective analysis is preferred as it is more true to life and necessitates that the model generalizes better than the holdout analysis. Both evaluation techniques show better results on the test set, which is relatively unusual. For the holdout analysis the difference is very small (about 0.02); this difference might simply be explained by the amount of data, in that the model evaluated on the validation set is trained using less data than the model evaluated on the test set. For the retrospective analysis where the difference is larger, the reason is probably similar: the previous version of DrugBank used for training during validation contains fewer

interactions (11,284 interactions) than the version used to train the final model which was used for testing (with 45,296 interactions).

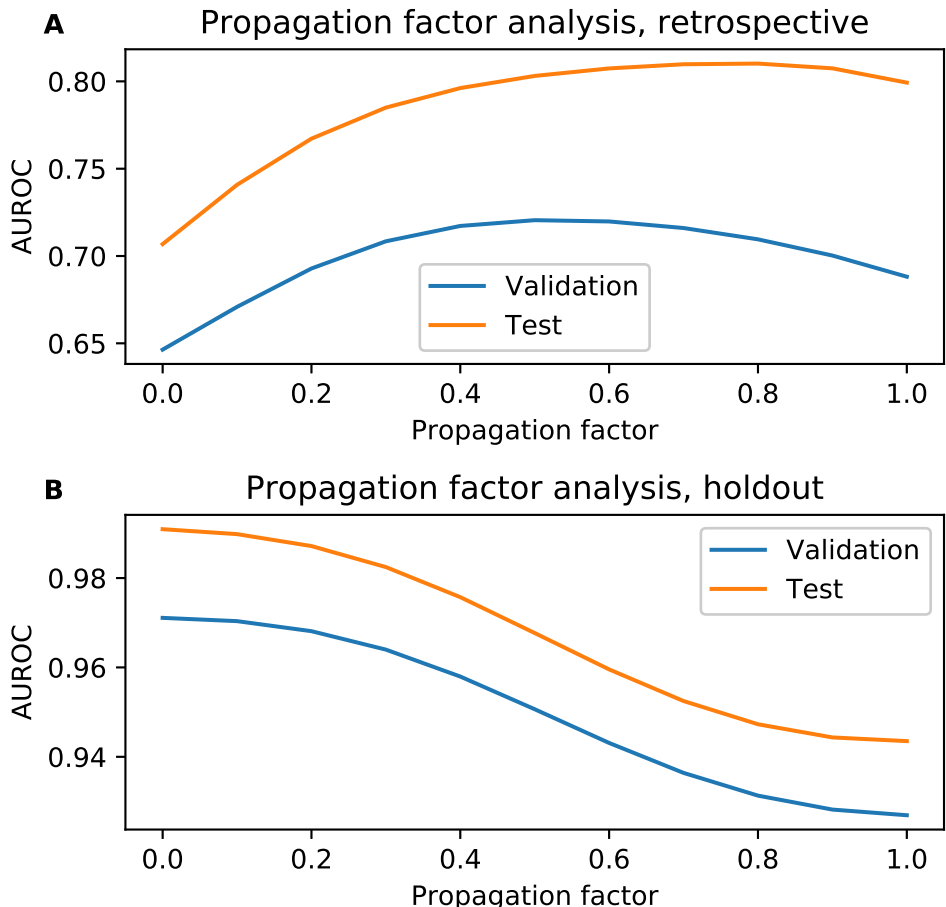


Fig 5. AMFP’s propagation factor analysis. **A)** Retrospective propagation factor analysis. The optimal value selected during validation and used for model training is 0.5. The optimal value for the test set is 0.8. **B)** Holdout propagation factor analysis. For both validation and training the optimal value is zero - weights are not propagated at all.

We report the optimal values for the parameters used: the embedding size used are 256 and 512 for the retrospective and holdout analysis respectively. For both evaluation schemes, the dropout is 0.3, and the learning rate is 0.01. It’s important to note that the dropout is applied separately on the embedding layer of each of the drugs. On average the number of entries in the embedding vector which are not affected at all by dropout during training is given by:

$$k \cdot (1 - p)^2, \quad (14)$$

where k is the embedding size, and p is the dropout ratio. Hence, for the retrospective analysis where the embedding size was 512 on average only 184.32 embedding entries are unaffected by the dropout during training. The optimal number of epochs used is five and six for the retrospective and holdout analysis respectively, and the batch sizes are 1024 and 256 respectively.

Some interesting observations can be made by comparing AMFP’s predictions regarding each drug pair with the pair’s structural similarity. One can expect that if two drugs share similar interactions it is likely that they have some structural similarity, however the predictions and structural similarity might be different or complementary. We computed the correlation coefficient between AMFP’s predictions and the structural similarity. For the structural similarity we used the method which was used by Ryu et al. [10]. This comparison showed a low correlation coefficient of 0.151 with $P - value < 10^{-15}$; for comparison, Vilar et al. [8] report a correlation coefficient of 0.167.

Practical contribution

The proposed models can be utilized to improve drug-drug interaction discovery and can be combined with additional structural information to improve drug-drug interaction detection performance.

In this paper, we present the top 100 predictions made by AMFP (see Supporting information section) after training it on release 5.1.1 of DrugBank and using the parameters optimized in the retrospective analysis; note that version 5.1.1 was the latest release available when our research was conducted. We manually validated the first 10 predictions made by AMFP; as of now, eight of them have been added to DrugBank. For the following five drug-drug interactions the metabolism of one drug can change due to the drug interaction: Curcumin and Primidone, Rifapentine and Fluvoxamine, Curcumin and Rifapentine, Lumacaftor and Fluvoxamine, and Curcumin and Lovastatin. For the following three drug-drug interactions the serum concentration of one drug can change due to the drug interaction: Ceritinib and Fluvoxamine, Curcumin and Clotrimazole, and Curcumin and Lumacaftor. No evidence was found for the existence of the following two drug-drug interactions: Pentobarbital and Sulfisoxazole, and Curcumin and Pentobarbital which might indicate two new unknown interactions predicted by our methods.

In addition, we created a list of the latent factors (embeddings) created for each drug and made this publicly available. These factors can be used as compressed representations of the drugs. The factors contain the structure of the interaction network and can provide a head start on downstream tasks in the form of transfer learning. For example, the drug embeddings created using the interaction network can be used to detect side effects.

AMF and AMFP can be scaled to support a large number of drugs and interactions; the models do not require training using all of the positive examples (existing interactions), and positive sampling can also be used, allowing the method to operate on very large datasets. Each of the proposed methods presented here required no more than a few minutes to train using a standard laptop. Hyperparameter optimization required more time, and this process can usually be executed efficiently by an expert; AutoML methods for parameter optimization are currently gaining interest, and such methods can dramatically reduce the time required for optimizing hyperparameters [30].

Conclusion

In this paper, we designed two methods for drug-drug interaction prediction based on a novel matrix factorization technique designed for adjacency matrices and developed useful in silico models to predict new drug interactions. Additionally, we train an XGBoost ensemble using various predictors. The methods were implemented and

made public, along with additional resources used in this research. Our methods were systematically validated through a retrospective and holdout evaluation using DrugBank (release 5.1.1 which contains 1,440 drugs and 248,146 drug-drug interactions), showing state of the art results with an area under receiver operating characteristic curve of 0.814 overall and accuracy of 56% when predicting the first interaction for each drug. Our methods can be used on a large-scale and applied for link prediction problems in domains other than drug-drug interaction prediction. Using the proposed DDI predictor, a database containing the most promising drug-drug interaction candidates is provided in the Supporting information section.

Supporting information

S1 Appendix. AMFP’s predictions: a list of 100 non-existing drug-drug interactions predicted by AMFP using a retrospective model on release 5.1.1 of DrugBank. The items are listed in descending order, based on the level of confidence. Curcumin and Primidone, Ceritinib and Fluvoxamine, Rifapentine and Fluvoxamine, Curcumin and Pentobarbital, Curcumin and Rifapentine, Lumacaftor and Fluvoxamine, Curcumin and Lovastatin, Pentobarbital and Sulfisoxazole, Curcumin and Clotrimazole, Curcumin and Lumacaftor, Primidone and Atomoxetine, Phenobarbital and Atomoxetine, Idelalisib and Fluvoxamine, Fluconazole and Fluvoxamine, Procaine and Methadone, Deferasirox and Primidone, Ceritinib and Clotrimazole, Stiripentol and Rifapentine, Stiripentol and Sulfisoxazole, Curcumin and Cobicistat, Cimetidine and Ziprasidone, Desipramine and Cyclosporine, Lumacaftor and Rifapentine, Curcumin and Pimecrolimus, Cobicistat and Pentobarbital, Nefazodone and Leflunomide, Curcumin and Fluconazole, Osimertinib and Sulfisoxazole, Pentobarbital and Atomoxetine, Ceritinib and Clemastine, Doxycycline and Fluvoxamine, Clemastine and Sulfisoxazole, Lumacaftor and Clotrimazole, Lumacaftor and Stiripentol, Stiripentol and Cobicistat, Curcumin and Clemastine, St. John’s Wort and Primidone, Quinidine and Tranylcypromine, Clemastine and Clotrimazole, St. John’s Wort and Fluvoxamine, Cobicistat and Sulfisoxazole, Deferasirox and Pentobarbital, Nefazodone and Torasemide, Sarilumab and Fluvoxamine, Idelalisib and Sulfisoxazole, Teriflunomide and Nefazodone, Selegiline and Quinidine, Acetyl sulfisoxazole and Pentobarbital, St. John’s Wort and Fosphenytoin, Selegiline and Azelastine, Ketoconazole and Tranylcypromine, St. John’s Wort and Phenobarbital, Curcumin and St. John’s Wort, Cobicistat and Clotrimazole, Rifapentine and Thioridazine, Desipramine and Isradipine, Nefazodone and Irbesartan, Osimertinib and Clotrimazole, Sarilumab and Primidone, Dasatinib and Desipramine, Lopinavir and Tranylcypromine, Rifapentine and Clemastine, Idelalisib and Clotrimazole, Clarithromycin and Metoprolol, Sarilumab and Phenobarbital, Ceritinib and Deferasirox, Deferasirox and Sulfisoxazole, Diphenhydramine and Verapamil, Luliconazole and Fluvoxamine, Tranylcypromine and Indinavir, Deferasirox and Rifapentine, Sarilumab and Fosphenytoin, Rotigotine and Escitalopram, Metoprolol and Sulfisoxazole, Lumacaftor and Cobicistat, Nilotinib and Tranylcypromine, Isoniazid and Ziprasidone, Thioridazine and Rifabutin, Lumacaftor and Clemastine, Simeprevir and Sulfisoxazole, Clotrimazole and Fluconazole, Curcumin and Doxycycline, Venlafaxine and Torasemide, St. John’s Wort and Pentobarbital, Nefazodone and Valsartan, Teriflunomide and Phenobarbital, Teriflunomide and Venlafaxine, Stiripentol and Fluconazole, Osimertinib and Clemastine, Darunavir and Tranylcypromine, Desipramine and Conivaptan, Itraconazole and Desipramine, Cobicistat and Clemastine, Desipramine and Rifabutin, Nefazodone and Ciprofloxacin, Rifampicin and Paroxetine, Phenytoin and Betaxolol, Vemurafenib and Nabilone, St. John’s Wort and Sulfisoxazole, Rifapentine and

References

1. Preventable Adverse Drug Reactions: A Focus on Drug Interactions;. Available from: <https://www.fda.gov/Drugs/DevelopmentApprovalProcess/DevelopmentResources/>
2. DW B, N S, DJ C, et al. The costs of adverse drug events in hospitalized patients. *JAMA*. 1997;277(4):307–311. doi:10.1001/jama.1997.03540280045032.
3. Raschetti R, Morgutti M, Menniti-Ippolito F, Belisari A, Rossignoli A, Longhini P, et al. Suspected adverse drug events requiring emergency department visits or hospital admissions. *Eur J Clin Pharmacol*. 1999;54(12):959–963.
4. Budnitz DS, Pollock DA, Weidenbach KN, Mendelsohn AB, Schroeder TJ, Annest JL. National surveillance of emergency department visits for outpatient adverse drug events. *JAMA*. 2006;296(15):1858–1866.
5. Qato DM, Wilder J, Schumm LP, Gillet V, Alexander GC. Changes in Prescription and Over-the-Counter Medication and Dietary Supplement Use Among Older Adults in the United States, 2005 vs 2011. *JAMA Intern Med*. 2016;176(4):473–482.
6. Guiding Principles for the Care of Older Adults with Multimorbidity: An Approach for Clinicians. *Journal of the American Geriatrics Society*;60(10):E1–E25. doi:10.1111/j.1532-5415.2012.04188.x.
7. Corrigan OP. A risky business: the detection of adverse drug reactions in clinical trials and post-marketing exercises. *Social Science & Medicine*. 2002;55(3):497 – 507. doi:[https://doi.org/10.1016/S0277-9536\(01\)00183-6](https://doi.org/10.1016/S0277-9536(01)00183-6).
8. Vilar S, Uriarte E, Santana L, Tatonetti NP, Friedman C. Detection of Drug-Drug Interactions by Modeling Interaction Profile Fingerprints. *PLOS ONE*. 2013;8(3):1–11. doi:10.1371/journal.pone.0058321.
9. Zhang P, Wang F, Hu J, Sorrentino R. Label Propagation Prediction of Drug-Drug Interactions Based on Clinical Side Effects. *Scientific Reports*. 2015;5:12339 EP –.
10. Ryu JY, Kim HU, Lee SY. Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the National Academy of Sciences*. 2018;115(18):E4304–E4311. doi:10.1073/pnas.1803294115.
11. Gottlieb A, Stein GY, Oron Y, Ruppim E, Sharan R. INDI: a computational framework for inferring drug interactions and their associated recommendations. *Molecular Systems Biology*. 2012;8(1). doi:10.1038/msb.2012.26.
12. Zhang W, Chen Y, Liu F, Luo F, Tian G, Li X. Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data. *BMC Bioinformatics*. 2017;18(1):18. doi:10.1186/s12859-016-1415-9.
13. Park K, Kim D, Ha S, Lee D. Predicting Pharmacodynamic Drug-Drug Interactions through Signaling Propagation Interference on Protein-Protein Interaction Networks. *PLOS ONE*. 2015;10(10):1–13. doi:10.1371/journal.pone.0140816.

14. Sangrak Lim KL, Kang J. Drug drug interaction extraction from the literature using a recursive neural network. *PLOS One*. 2018;13(1):e0190926. doi:10.1371/journal.pone.0190926.
15. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS. Neural Collaborative Filtering. In: *Proceedings of the 26th International Conference on World Wide Web. WWW '17*. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee; 2017. p. 173–182. Available from: <https://doi.org/10.1145/3038912.3052569>.
16. Wu H, Zhang Z, Yue K, Zhang B, He J, Sun L. Dual-regularized matrix factorization with deep neural networks for recommender systems. *Knowledge-Based Systems*. 2018;145:46 – 58. doi:<https://doi.org/10.1016/j.knosys.2018.01.003>.
17. Fan J, Cheng J. Matrix completion by deep matrix factorization. *Neural Networks*. 2018;98:34 – 41. doi:<https://doi.org/10.1016/j.neunet.2017.10.007>.
18. Diederik P Kingma JB. Adam: A Method for Stochastic Optimization. In: *International Conference on Learning Representations (ICLR)*; 2015.
19. Cukierski W, Hamner B, Yang B. Graph-based features for supervised link prediction. In: *The 2011 International Joint Conference on Neural Networks*; 2011. p. 1237–1244.
20. Adamic LA, Adar E. Friends and neighbors on the Web. *Social Networks*. 2003;25(3):211 – 230. doi:[https://doi.org/10.1016/S0378-8733\(03\)00009-1](https://doi.org/10.1016/S0378-8733(03)00009-1).
21. Chen H, Li X, Huang Z. Link prediction approach to collaborative filtering. In: *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05)*; 2005. p. 141–142.
22. Fire M, Tenenboim-Chekina L, Puzis R, Lesser O, Rokach L, Elovici Y. Computationally Efficient Link Prediction in a Variety of Social Networks. *ACM Trans Intell Syst Technol*. 2014;5(1):10:1–10:25. doi:10.1145/2542182.2542192.
23. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16*. New York, NY, USA: ACM; 2016. p. 785–794. Available from: <http://doi.acm.org/10.1145/2939672.2939785>.
24. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res*. 2018;46(D1):D1074–D1082.
25. Yang Y, Lichtenwalter RN, Chawla NV. Evaluating link prediction methods. *Knowledge and Information Systems*. 2015;45(3):751–782. doi:10.1007/s10115-014-0789-0.
26. Chollet F, et al.. Keras; 2015. <https://keras.io>.
27. Hagberg A, Swart P, S Chult D. Exploring network structure, dynamics, and function using networkx. *Proceedings of the 7th Python in Science Conference (SciPy 2008)*. 2008;.
28. Sun X, Xu W. Fast Implementation of DeLong’s Algorithm for Comparing the Areas Under Correlated Receiver Operating Characteristic Curves. *IEEE Signal Processing Letters*. 2014;21(11):1389–1393. doi:10.1109/LSP.2014.2337313.

29. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterton M, editors. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. vol. 9 of Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR; 2010. p. 249–256. Available from: <http://proceedings.mlr.press/v9/glorot10a.html>.
30. Sun Y, Xue B, Zhang M, Yen GG. An Experimental Study on Hyper-parameter Optimization for Stacked Auto-Encoders. In: 2018 IEEE Congress on Evolutionary Computation (CEC); 2018. p. 1–8.