Project Final

Anushree Dhople and Gazi Muhammad Samiul Hoque
{adhople2, ghoque2}@illinois.edu

Group ID: **24**
Paper ID: **208**, Difficulty: **hard**
Presentation link: https://mediaspace.illinois.edu/media/t/1_z467duu0
Code link: https://github.com/samhq/cs598dl4h-project

1 Introduction

Drug recommendation is a critical step in patient care. It usually involves knowing the current as well as the longitudinal history of the patient, patient demographics such as age, drug-drug interactions and side effects. With the availability of massive medical records in the EHR, AI models can be applied to generate better drug recommendations especially for critical patients or those with co-morbidities.

The current drug recommendation process suffers from numerous problems such as consideration of insufficient patient history. To overcome this, the entire EHR history of the patient gathered at visit time stamps can be taken into account to generate the predictive model.

Other problems are around the incorporation of patient demographics along with existing DDI models that use indirect constraints such as knowledge graphs. Drugs are usually encoded as one-hot vectors ignoring the molecular structure of the drug which could yield critical side effects and drug properties.

In **SafeDrug** (Yang et al., 2021) the authors have considered DDI interactions explicitly and leveraged the molecular structure and substructures of the drug's molecule to arrive at a safe drug recommendation.

SafeDrug achieves this by first learning the entire patient history from the EHR data and creates a patient representation. The patient representation is fed into a dual molecular graph encoder to capture the global pharmacological properties as well as the substructures of the drug. In this way both the longitudinal patient data as well as the drug's molecular properties are considered. A threshold is applied to the output of the dual molecular graph encoders to generate the final drug recommendation list.

2 Scope of reproducibility

This paper focuses on constructing a comprehensive model for safe drug recommendation while reducing drug-drug interactions by taking into consideration longitudinal patient representation, drug molecular structure to learn meaningful molecular drug properties such as efficacy and safety profiles, as well as introducing a symmetric binary adjacency matrix D to denote the DDI relation.

SafeDrug outperforms the baseline models of LR, ECC, RETAIN, LEAP, DMNC and GAMENet in significantly lower drug-drug interaction rates and higher accuracy.

2.1 Addressed claims from the original paper

In our selected paper, the authors claimed that their proposed model could achieve *lower* DDI score than other baseline models, such as LR, ECC, RETAIN, LEAP, DMNC and GAMENet. SafeDrug also achieves *higher* Jaccard coefficient, *higher* F1-score and *higher* PRAUC values. We tested these claims and found the following in our initial run (detailed in Table 4):

- SafeDrug achieved a **lower** DDI than LR, ECC, RETAIN, LEAP and GAMENet. A lower DDI in drug recommendations signifies better predictive power.
- SafeDrug achieved a better Jaccard coefficient than LR, ECC, RETAIN, LEAP and GAMENet. The Jaccard coefficient computes the similarity between the drug recommendation and their corresponding ground truths. A **higher** Jaccard coefficient signifies a more accurate drug recommendation.
- SafeDrug achieved a **higher** F1 score than LR, ECC, RETAIN, LEAP and GAMENet.

- SafeDrug achieved a **higher** PRAUC (Precision Recall Area Under Curve) than LR, ECC, RETAIN, LEAP and GAMENet.

3 Methodology

3.1 Model descriptions

The SafeDrug model comprises of 4 basic components. The first component is the patient representation module which learn patient demographics from the EHR data. A diagnosis embedding and a procedure embedding are created to store their respective patient data. In the diagnosis embedding each row corresponds to a diagnosis for the patient and the size of the row is dimension of the embedding space. A procedure embedding is similarly created. Two separate RNNs are then employed to obtain the hidden diagnosis and procedure vectors which are concatenated to create a compact patient representation vector.

$$\mathbf{h}^{(t)} = \mathbf{NN}_1([\mathbf{d}_h^{(t)} \# \mathbf{p}_h^{(t)})]; \mathbf{W}_1)$$

The second and third components of SafeDrug take the patient representation and generate global and local molecular structural embeddings respectively and in parallel.

To create the global molecular structural embedding, SafeDrug encodes drug molecule data using a MPNN model. The drug molecular graph can be naturally represented by MPNN. A MPNN is a specialized graph neural network (GNN). GNNs are a class of deep learning methods designed to perform inference on data described by graphs. Compared with other representations such as vectors or sequences, graphs are expressive models that can capture more complex interactions between heterogeneous biomedical concepts. Since a drug's molecular structure is naturally represented as a graph it is ideal to apply GNNs to their analysis. The graph vertices represent the atoms and the graph edges represent the atom-atom connection. Through the application of a GNN to a drug's molecular structure SafeDrug can learn meaningful molecular drug properties such as efficacy and safety profiles.

These embeddings are then collected into a drug memory where each row represents a single drug. Given a patient representation the objective is to obtain the most relevant drug from the drug memory. This is achieved by taking a dot product of the patient representation and the MPNN embedding and further applying sigmoid function.

$$\mathbf{m}_r^{(t)} = \sigma_2((\mathbf{E}_g)\mathbf{h}^{(t)})$$

The $\mathbf{m}_r^{(t)}$ vector stores the matching score for one drug. This score is further parameterized by a feed forward neural network,

$$\mathbf{m}_g^{(t)} = \mathbf{LN}(\mathbf{m}_r^{(t)} + \mathbf{NN}_2(\mathbf{m}_r^{(t)}; \mathbf{W}_2))$$

To create the local molecular structural embedding, SafeDrug captures the drug's local functionality by building a substructure-to-drug bipartite architecture. First, the drug molecule is broken up into substructures that preserve the most critical drug functional groups and bonds. Using this set of substructures S, SafeDrug creates a mask matrix H where $\mathbf{H}_{ij} = 1$, denotes that drug j has substructure i. Now, taking the patient representation $\mathbf{h}^{(t)}$ as input, it first applies a feedforward neural network and furthermore a sigmoid function to generate a local functional vector,

$$\mathbf{m}_f^{(t)} = \sigma_2(\mathbf{NN}_3(\mathbf{h}^{(t)}; \mathbf{W}_3))$$

This vector represents functionalities the drug to be prescribed to the patient should have. These functionalities are computed, as above, based on the disease profile of the patient. Now, the objective is to find drugs that cover or provide these functionalities while keeping DDI at its minimum. To do so, SafeDrug creates a 1-layer neural network where the parameter matrix $\mathbf{W}_4$ is masked by the sparse mask matrix H by a matrix element wise dot product. In the training stage, the neural network learns to map the patient representation vector $\mathbf{m}_f^{(t)}$ onto the corresponding local drug representation. Thus, first the model has fewer parameters as H as sparse and second, DDI is avoided.

$$\mathbf{m}_l^{(t)} = \mathbf{NN}_4(\mathbf{m}_f^{(t)}; \mathbf{W}_4 \odot \mathbf{H})$$

The final drug recommendation is now generated by performing an element wise matrix multiplication between the global drug matching vector and the local drug representation. A sigmoid function is further applied to scale the output. By applied a threshold of δ, the list of drug recommendations can be generated.

$$\hat{\mathbf{o}}^{(t)} = \sigma_3(\mathbf{m}_g^{(t)} \odot \mathbf{m}_l^{(t)})$$

We listed the parameter count for each model in our initial run in Table 1.

Model	Parameter Count
RETAIN	285489
LEAP	177395
GAMENet	444209
SafeDrug	368777

Table 1: Number of parameters for the Models

3.2 Data descriptions

This paper uses the MIMIC III dataset (Johnson et al., 2016b,a) which is publicly available on request from PhysioNet[1]. We got the access and downloaded the dataset from PhysioNet. We needed three files from this dataset, *PRESCRIPTIONS.csv*, *DIAGNOSES_ICD.csv* and *PROCEDURES_ICD.csv*. We also needed a couple of files for the drugs mapping, such as RXCUI to ATC4 mapping, CID code to ATC mapping, rxnorm to RXCUI mapping, drug information table to map drug name to drug SMILES string, and CID coded drug DDI information mapping. Details on their usage are given in our code repository. The statistics of the data after the pre-processing is reported in the following Table 2.

Type of Data	Size
# of Patients	6350
# of clinical events (visits)	15032
avg # of of visits	2.367
max # of visits of a patient	29
# of diagnoses	1958
avg # of diagnoses	10.509
max # of of diagnoses of a patient	128
# of medicines	112
avg # of medicines	11.648
max # of of medicines of a patient	64
# of procedures	1430
avg # of of procedures	3.844
max # of procedures of a patient	50

Table 2: Data Statistics

3.3 Hyperparameters

The hyperparameters of SafeDrug were set as stated in the paper as follows: threshold δ is 0.5, the weight $\alpha = 0.95$, $K_p = 0.05$, and acceptance rate γ is selected 0.06.

3.4 Implementation

We have copied the existing code contained in the GitHub repository **ycq091044/SafeDrug**[2] into our own repository at **samhq/cs598dl4h-project**[3]. We modified the models initialization for easier access to the parameters for each model. We have also added some scripts to run the training and inference in background for long running jobs. We modified the codes for further testing of different settings and ablation testing.

3.5 Computational requirements

We ran the models in a Google Cloud Deep Learning VM[4] with 4 vCPU and 26GB of RAM. It has a NVIDIA Tesla K80 GPU with 12 GB GDDR5 memory. It has the PyTorch:1.11 (PyTorch/CUDA11.3.GPU) framework installed by the cloud team by default. The total execution time for each models are given in Table 3:

Model	Epoch	Total Run-time
LR	1	361
ECC	1	3408
RETAIN	50	1498
LEAP	50	17682
GAMENet	50	5417
SafeDrug	50	20628
TOTAL		48994

Table 3: Total run-time (training and inference) for each model (in $seconds$ - wall clock time)

We continued testing the models with different parameters and different settings to check for performance improvements. We modified the scripts to get more metrics from the execution such as max CPU/GPU memory usage and found out that the SafeDrug model took a max of 670MB GPU memory during training phase.

4 Results

The experiments were carried out on the MIMIC III dataset. We computed results on the following 5 efficacy metrics: DDI, Jaccard, F1-score, PRAUC and Avg. # of Drugs. In our results we compared SafeDrug with the following baselines: LR (standard logistics regression), ECC (Ensemble Classifier Chain) (Read et al., 2009), RETAIN (Choi et al., 2016), LEAP (Zhang et al., 2017) and GAMENet (Shang et al., 2019).

From Table 4, we can observe that SafeDrug outperforms all other baseline methods with lower DDI and higher accuracy. The efficacy metrics

[1] https://physionet.org/

[2] https://github.com/ycq091044/SafeDrug

[3] https://github.com/samhq/cs598dl4h-project

[4] https://cloud.google.com/deep-learning-vm/docs/introduction

Model	DDI	Jaccard	F1-score	PRAUC	Avg. # of Drugs
LR	0.0775	0.4900	0.6470	0.7553	-
ECC	0.0806	0.4868	0.6428	0.7602	-
RETAIN	0.0851 ± 0.0028	0.4711 ± 0.0140	0.6337 ± 0.0129	0.7512 ± 0.0126	17.9925 ± 0.8751
LEAP	0.0689 ± 0.0028	0.4369 ± 0.0117	0.6002 ± 0.0116	0.6467 ± 0.0068	19.1096 ± 0.1240
GAMENet	0.0836 ± 0.0067	0.4790 ± 0.0260	0.6382 ± 0.0240	0.7393 ± 0.0247	25.1478 ± 1.1325
SafeDrug	0.0627 ± 0.0023	0.5051 ± 0.0150	0.6624 ± 0.0134	0.7604 ± 0.0117	19.3245 ± 0.5557
SafeDrug*	0.0589 ± 0.0005	0.5213 ± 0.0030	0.6768 ± 0.0027	0.7647 ± 0.0025	19.9178 ± 0.1604

** values from the original SafeDrug model paper*

Table 4: Statistics from initial run of all models and comparison from the data from the paper

from the original SafeDrug model paper are also added for comparison and reference.

4.1 Result 1

Our SafeDrug model achieves a lower DDI than all other baseline models. We reproduced the DDI to within 6.4% of the reported value in the original paper.

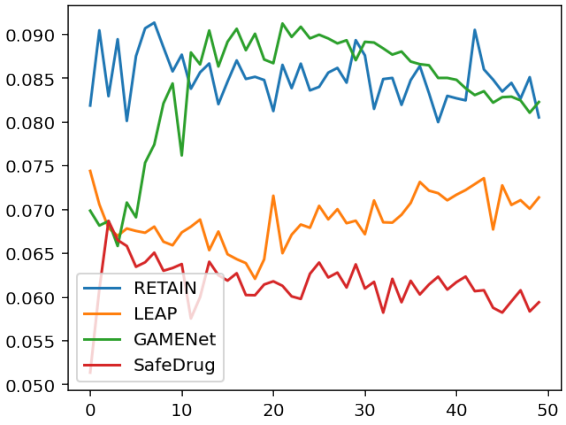

Figure 1: DDI Rate changes over epochs

From Figure 1, we verified the claim that the SafeDrug model did achieve lower DDI than all models in all epochs.

4.2 Result 2

Our model achieved higher Jaccard than all other baseline models. We reproduced Jaccard to within 3% of the reported value in the original paper.

4.3 Results 3

Our model achieved a higher F1-score than all other baseline models. We reproduced the F1-score to within 2% of the reported value in the original paper.

4.4 Results 4

Our model achieved a higher PRAUC than all other baseline models. We reproduced the F1-score to within 1% of the reported value in the original paper.

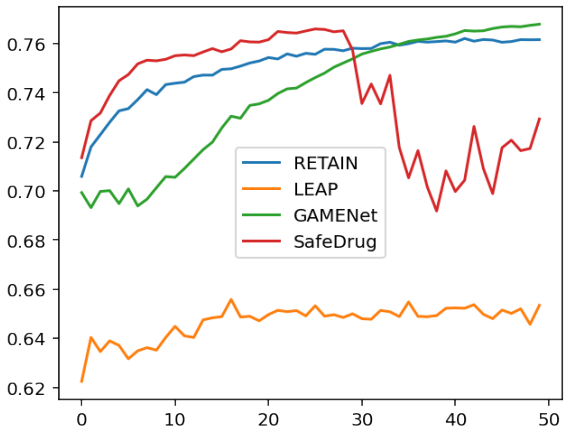

Figure 2: PRAUC changes over epochs

4.5 Additional results not present in the original paper

We completed a successful run of the stated baseline models as demonstrated in Table 4.

Furthermore, we conducted ablation studies to change various hyper parameters and measure the accuracy of the model. We modified the DDI rate, learning rate and the number of epochs to measure their impact on the prediction accuracy. These results are documented in Table 5.

5 Discussion

The original paper was reproducible to a large degree. From the basline results documented in Table 4 we can see that our code reproduced the DDI to within 6.45% of the original SafeDrug results as published in the paper. We were also able to create the baseline models of LR, ECC, RETAIN, LEAP and GAMENet. The only baseline model we faced issues in coding was the DMNC (Le et al., 2018) due to conflicting libraries.

We were also able to conduct ablation studies over a range of target ddi γ values as well as the

	LR	γ	DDI	Jaccard	F1-score	PRAUC	Avg. # of Drugs
SafeDrug	5e-04	0.06	0.0615 ± 0.0026	0.4851 ± 0.0212	0.6403 ± 0.0235	0.7396 ± 0.0226	19.0607 ± 0.5732
Ablation 1	0.01	0.02	0.0240 ± 0.0071	0.4211 ± 0.0071	0.5844 ± 0.0064	0.6793 ± 0.0123	19.3800 ± 2.0774
Ablation 2	0.001	0.03	0.0447 ± 0.0032	0.4687 ± 0.0061	0.6294 ± 0.0056	0.7282 ± 0.0039	17.4044 ± 0.6018
Ablation 3	1e-04	0.04	0.0466 ± 0.0031	0.4440 ± 0.0126	0.6036 ± 0.0151	0.6997 ± 0.0230	17.1075 ± 1.0902
Ablation 4	1e-05	0.05	0.0494 ± 0.0077	0.4184 ± 0.0071	0.5821 ± 0.0073	0.6980 ± 0.0071	18.7053 ± 1.6778
Ablation 5	1e-06	0.06	0.0656 ± 0.0055	0.4110 ± 0.0141	0.5745 ± 0.0148	0.6668 ± 0.0376	19.6280 ± 1.3323

Table 5: Statistics from ablation testing

learning rate to further support the claims made in the paper.

Analysis of change in target DDI. We evaluated how well the target DDI can be used to control the average DDI rate of our model's output. In SafeDrug the target DDI (γ) is used to control the loss function. The controllable factor β,

$$\beta = \begin{cases} 0, & \text{DDI} \leq \gamma \\ \max\left(0, 1 - \frac{\text{DDI}-\gamma}{K_p}\right), & \text{DDI} > \gamma \end{cases}$$

The key of the SafeDrug model is to provide as a low a DDI rate as possible to ensure the safety of the recommended drug combinations. To control this, a target DDI rate is used. If the model's DDI rate is less than the target DDI rate, then we will only consider maximizing the prediction accuracy. Else, β is adjusted to reduce the DDI as well.

From the results documented in Table 5 it can be observed that the DDI rate can be controlled by modifying γ. We have tested for γ values ranging from 0.02 to 0.06. For each γ, we trained a separate model. We ran one study for each value. In our study, the γ does not serve as the upper bound of the output DDI rate but it shows a correlation. As the γ is increased the output DDI rate also increases. We have confidence that given more time, we could have fine tuned the model to ensure γ served as the upper bound for the output DDI rate. It should also be noted that as γ becomes larger, more drugs are allowed in one combination. When γ is low, the model accuracy drops.

Analysis of change in learning rate. Deep learning models are typically trained by a stochastic gradient descent optimizer. SafeDrug uses Adam. In our ablations we have modified the learning rate. The learning rate parameter tells the optimizer how far to move the weights in the direction opposite of the gradient for a mini-batch. If the learning rate is low, the training is more reliable, but optimization will take a lot of time because the steps taken towards the min of the loss function are tiny. If the learning rate is high, then training may not converge. Weight changes can be so big that the optimizer overshoots the minimum and makes the loss worse.

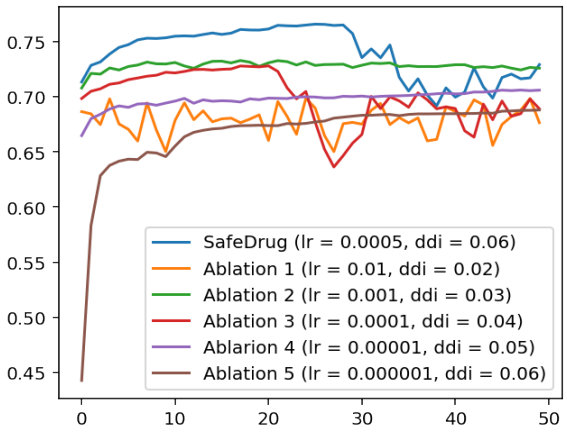

Figure 3: PRAUC changes in each ablation testing

We conducted studies for learning rates from 0.01 to 1e-06 decreasing in factors of 1e-01. From the results in Table 5 and in Figure 3, it can be seen that our results are inconclusive as far as modifications to learning rate is concerned. A possible reason for this could be of how we structured the ablation runs. It would have been more fruitful to separate out the ablation runs for modifications in learning rate from the ablation runs for modifications in γ. In our current results, as stated in Table 5, the accuracy values are worst for large learning rate value of 0.01. This is to be expected, as the higher learning rate does not allow the model to converge and arrive at a optimum solution. However, for significantly lower learning rate we should have noted better accuracy values. However the accuracy values significantly decrease for learning rates from 0.001 to 1e-06. This is also expected, since in the same experiments we have modified the γ values.

5.1 What was easy

The SafeDrug model was very well described in the original paper. All the sub models are well explained with clear explanations of how they are

generated. For example, to generate the patient representation the paper describes in detail how the diagnosis and procedure embedding are created by inputting patient data from the csv files. The paper further details how separate RNNs are used on these embeddings to obtain the diagnosis and procedure vectors which are further concatenated to create a compact patient representation. It was easy to follow the models in the code and to map how they arrive at the final recommendation.

The paper also provided exhaustive implementation details that clarified how the dataset was split into training, validation and test data as $\frac{2}{3} : \frac{1}{6} : \frac{1}{6}$. All parameters for each model were well documented with explanations where applicable. For example, choice of γ as 0.06 is well explained. However, justification of $\mathbf{K}_p$ as 0.05 is not given.

In the end, the data was also easily available by completing the required trainings and applying for it well in advance.

5.2 What was difficult

We had to spend a significant amount of time in debugging the problem with the DMNC baseline model. It needed the dnc package to run successfully, but no published version (we tried all from version 0.0.1 to 1.1.0) worked with this model.

Overall, the baseline results could be executed with some effort. However, each run with 50 epoch of the model plus baseline results takes between 7-8 hours. We had to ration our resources and usage of GPU in Google Cloud Platform to cut down on the ablation studies we could run. Our final run with all models and 5 ablation studies took around 41 hours to finish in the mentioned cloud VM in 3.5.

Each run requires significant resources to verify and while it would have been more fruitful to run more exhaustive ablation studies we were limited by the availability of computing resources available to us.

5.3 Recommendations for reproducibility

SafeDrug documents each sub model exhaustively and the hyper parameters for each model including the number of iterations run are well provided. However, it would be useful to get a better view of the patient representation used. This seems to be the key to how drug recommendations are generations and modifications to it will provide significant variations in the drug recommendation process. The other input to the model are the drug molecules which are again well documented and not much can be done by the end user in modifying those.

The paper notes that the patient representation is generated from the patient diagnosis, procedure and medication information available in DIAGNOSIS_ICD.csv, PROCEDURES_ICD.csv and PRESCRIPTIONS_ICD.csv. It can be recommended that to improve reproducibility the authors detail what values are exracted from these tables to construct the multi-hot vectors. This will allow end users, to also easily apply more ablations by modifying the inputs to the multi-hot vectors.

Further, choice of some hyperparameters is not well explained. For example, in computing β the correcting factor $\mathbf{K}_p$ is taken as 0.05. While reasonable justification can be found for other parameters it is not clear how this parameter was taken and if experimental results were conducted to arrive at an optimum value.

6 Communication with original authors

We did not communicated with the original authors yet. We have a plan to send this completed report to get their feedback and assessments. According to their feedback, we might try different approaches in the future.

References

Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3512–3520, Red Hook, NY, USA. Curran Associates Inc.

Alistair Johnson, Tom Pollard, and Roger Mark. 2016a. "mimic-iii clinical database" (version 1.4). *PhysioNet*.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016b. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035.

Hung Le, Truyen Tran, and Svetha Venkatesh. 2018. Dual memory neural computer for asynchronous two-view sequential learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1637–1645. ACM.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multi-label classification. In *Proceedings of the 2009th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECMLPKDD'09, page 254–269, Berlin, Heidelberg. Springer-Verlag.

Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2019. Gamenet: Graph augmented memory networks for recommending medication combination. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. 2021. Safedrug: Dual molecular graph encoders for recommending effective and safe drug combinations. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3735–3741. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yutao Zhang, Robert Chen, Jie Tang, Walter F. Stewart, and Jimeng Sun. 2017. Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1315–1324, New York, NY, USA. Association for Computing Machinery.