Final Report: Reproducibility of SafeDrug Model for Drug Recommendation

Chen Yuan and Xuan Chen
{cheny9, xuanc4}@illinois.edu

Group ID: 6
Paper ID: 208, Difficulty: [hard]
Presentation link: https://mediaspace.illinois.edu/media/t/1_r79jms0p
Code link: https://github.com/ricaelum42/Replication-of-SafeDrug

1 Introduction

To address the limitation of the instance-based medication recommendation models, longitudinal-based models are introduced. However, there exist other limitations: (1) Existing works ignore the molecular structure of drug properties as medications are represented using one-hot encoding that simplifies meaningful graphs into binary units. (2) Some existing works model Drug-Drug Interaction (DDI) via soft or indirect constraints, which result in non-controllable rates or compromised accuracy in the final recommendation. This paper proposes a new model, SafeDrug, which learns both patient representation and molecule representation and explicitly models DDI. (Yang et al., 2021)

2 Scope of reproducibility

The paper claims that the SafeDrug model can capture and leverage complex medical information from molecular structures in drugs. As a result, when compared with existing approaches such as Logistic Regression models (LR), Ensemble Classifier Chain (ECC), Deep Learning Models like RETAIN, LEAP, DMNC and GAMENet; SafeDrug medication recommendation has a reduced DDI rate and an increased Jaccard similarity score between the model proposed and the actual drug prescription.

We will replicate the SafeDrug model and use LR, RETAIN(Choi et al., 2016), LEAP(Zhang et al., 2017) and GAMENet(Shang et al., 2018) as the baseline models. Next, we will compare the outcomes of the SafeDrug model against the baseline models' to validate the claims by the author.

2.1 Addressed claims from the original paper

In this project, we will test the following claims made in the paper:

- Leveraging drug molecule structures will reduce DDI on the final drug recommendation and have improved accuracy. (Yang et al., 2021)
- The SafeDrug model consists of fewer parameters when compared to existing models and significantly improves the training speed. (Yang et al., 2021)

3 Methodology

We aim to use the author's code for the purpose of replication. Our resources and references include:

- Code from the original paper: https://github.com/ycq091044/SafeDrug
- Code from baseline model (GAMENet) paper: https://github.com/sjy1203/GAMENet
- Google Colab Pro

3.1 Model descriptions

The SafeDrug model focus on leveraging Electronic Health Records (EHR) data and drug moledular structures to make drug recommendation with a reduced DDI rate. The SafeDrug model consists of four components:

1. **Longitudinal Patient Representation Module.** The objective of this module is to learn patient representation from diagnosis and procedure information. It consists of two embedding vectors generated by RNN models: A diagnosis embedding and a procedure embedding. The longitudinal patient representation module aims to learn the dynamic patient health condition and therefore can to distinguish between a newly diagnosed disease and a chronic disease. (Yang et al., 2021)

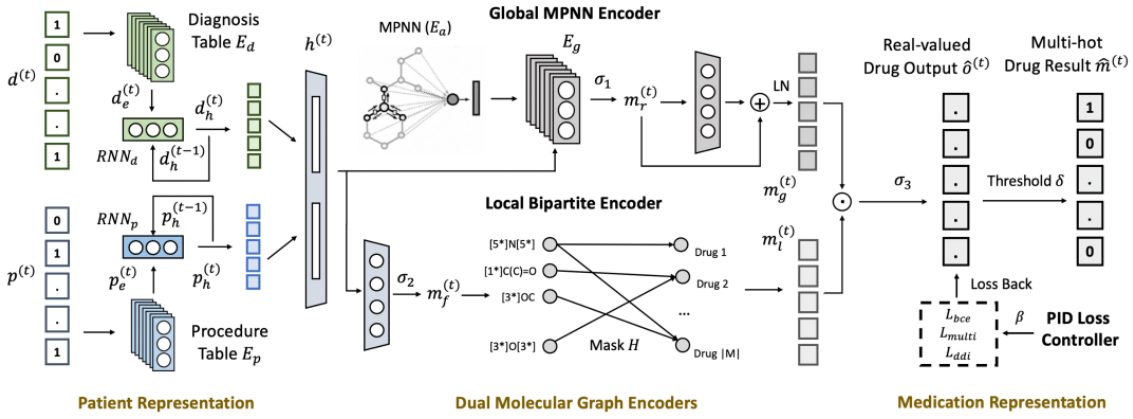

Figure 1: Structure of SafeDrug (Yang et al., 2021)

2. **Global Molecular Graph Encoder.** An MPNN encoder to generate a global drug vector which measures the similarity between patient representation and drug representation. (Yang et al., 2021)

3. **Local Molecular Graph Encoder.** A bipartite encoder to generate a local drug vector encoding the molecular properties of a drug. (Yang et al., 2021)

4. **Medication Representation.** This step simply combines the global drug vector and the local drug vector element-wise. It is basically using the global drug vector as attention weights to adjust the local drug vector. The output gives the drug recommendation. (Yang et al., 2021)

The `SafeDrug` model uses a DDI controllable loss function, which has two components:

- A loss function for the multi-label classification task. This model uses a binary cross-entropy (BCE) loss function combined with a multi-label hinge loss function.

  $$L_{BCE} = -\sum_{i=1}^{|M|} m_i^{(t)} log(\hat{o}_i^{(t)}) + (1 - m_i^{(t)}) log(1 - \hat{o}_i^{(t)})$$

  $$L_{multi} = \sum_{i,j:m_i^{(t)}=1,m_j^{(t)}=0} \frac{max(0, 1-(\hat{o}_i^{(t)} - \hat{o}_j^{(t)}))}{|M|}$$

- An adverse DDI loss function to control the DDI rate.

  $$L_{ddi} = \sum_{i=1}^{|M|} \sum_{j=1}^{|M|} D_{ij} \cdot \hat{o}_i^{(t)} \cdot \hat{o}_j^{(t)}$$

  where $|M|$ is the number of drugs.

The final loss function is a weighted average of the above components. To control the DDI rate, the author uses a threshold γ to control the maximum allowed DDI loss rate. If the DDI loss rate is above the threshold, then the `SafeDrug` model will adjust the weights of the two loss function components to achieve a lower DDI (Yang et al., 2021).

$$L = \beta(\alpha L_{BCE} + (1-\alpha) L_{multi}) + (1-\beta) L_{ddi}$$

β will be adjusted to give more weights to DDI loss function if DDI is above γ.

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

where α is a pre-defined hyperparameter and K_p is the correcting factor.

3.2 Data descriptions

The replication work uses the same datasets as in the paper:

- MIMIC-III patient information (Johnson et al., 2016), can be obtained from `https://physionet.org/content/mimiciii/1.4/`. This dataset consists of 40,000+ patients who stayed in the ICU of the Beth Israel Deaconess Medical Center from 2001 through 2012. In this paper, patients with only one visit were excluded from the base.

- Drug info that can be downloaded from `drugbank.com`.

- Other prescription, diagnosis, procedure and DDI information comes directly from GitHub repository of the original paper.

The data were preprocessed using the same way as the original paper. Data statistics are shown in Table 1.
The training data, validation data and testing data corresponds to 2/3, 1/6 and 1/6 of the input dataset.

Table 1: Input Data Statistics

Statistic	Value
Tot # of patients	6,350
Tot # of clinical events	15,032
Tot # of diagnosis	1,958
Tot # of procedures	1,430
avg # of diagnosis	10.5
avg # of medicines	11.6
avg # of procedures	3.8

3.3 Hyperparameters

We used the same set of hyperparameters as provided in the paper. The values of the hyperparameters are shown in Table 2.

Table 2: Hyperparameters (Yang et al., 2021)

Name	Value
Epoches	50
Learning Rate	2×10^{-4}
target DDI γ	0.06
K_p	0.05
α	0.95
threshold δ	0.5

3.4 Implementation

We used the existing code for replication. The existing code consists of both the benchmark models and the `SafeDrug` model proposed by the authors. The link to our replication work is as follows:

- https://github.com/ricaelum42/Replication-of-SafeDrug

3.5 Computational requirements

In this experiment, we used Google Colab Pro as our main computational resource. Google Colab Pro provides the Tesla K80 GPU with 13G Memory Space. Prior to the experiment, we were thinking of using the free version, however later on we decided to upgrade to the pro version, which reduced the execution time by almost half.

4 Results

We have completed one run for the `SafeDrug` model using the source code provided by the author. Our analysis shows that we were able to reproduce the DDI rate within 5% of the reported value and the Jaccard score within 2% of the reported value. We've also replicated four baseline models (LR, RETAIN, LEAP, and GAMENet) to compare with `SafeDrug`, the results are documented in Table 4. Based on the performance from our replicated models, we can conclude and verify that `SafeDrug` outperforms the baseline models with a lower DDI rate and better accuracy. Note that in the original paper, the testing was conducted by taking the average of 10 runs based on bootstrap sampling on the testing data. Due to limitation in time and computational resources, we only calculated one number based on the entire testing set. However, the conclusion still holds based on our calculation. Based on the training time, we have sufficient evidence to prove that `SafeDrug` is more computationally efficient compared with LEAP. However, since Colab is a shared computation resource, the model training time can highly depend on the time of the day and overall website traffic. Therefore, we do not have enough evidence to show that `SafeDrug` is computationally more effective than GAMENet.

4.1 Performance Assessment of `SafeDrug`

We compared our replicated model performance (Table 4) with the performance provided in the paper (Table 3). We also compared our replicated `SafeDrug` model performance with the performance of replicated baseline models. Our replicated model performance supports the claim by the author that `SafeDrug` model has a reduced DDI rate and enhanced Jaccard score compared with baseline models.

Table 3: Model Performance in the original paper (Yang et al., 2021)

Model	DDI	Jaccard
LR	0.0829 ± 0.0009	0.4865 ± 0.0021
RETAIN	0.0835 ± 0.0020	0.4887 ± 0.0028
LEAP	0.0731 ± 0.0008	0.4521 ± 0.0024
GAMENet	0.0864 ± 0.0006	0.5067 ± 0.0025
`SafeDrug`	0.0589 ± 0.0005	0.5213 ± 0.0030

Table 4: Replicated Model Performance

Model	DDI	Jaccard
LR	0.0777	0.4903
RETAIN	0.0800	0.4857
LEAP	0.0729	0.4496
GAMENet	0.0751	0.5145
SafeDrug	0.0619	0.5111

4.2 Model Efficiency of SafeDrug

In addition, we also documented the model efficiency of the runs (Table 6). Due to the difference in computational resources, we are not compare our model efficiency directly with the one in the original paper. However, based on our replication, we can conclude that the SafeDrug model is computationally more efficient compared with LEAP.

Table 5: Model Efficiency in the original paper (Yang et al., 2021)

Model	# of Param.	Train Time	Test Time
RETAIN	287,940	36.35s /Epoch	3.98
LEAP	433,286	336.14s /Epoch	32.31
GAMENet	449,092	162.10s /Epoch	26.85
SafeDrug	325,473	138.77s /Epoch	10.64s

Table 6: Replicated Model Efficiency

Model	# of Param.	Train Time	Test Time
RETAIN	285,489	37.41 /Epoch	7.73s
LEAP	428,403	278.32s /Epoch	29.21s
GAMENet	444,209	114.25s /Epoch	15.07s
SafeDrug	366,122	239.16s /Epoch	11.77s

5 Discussion

We think the original paper is highly reproducible given that the source code for SafeDrug and its baseline models are publicly available. We were able to successfully reproduce the performance results of SafeDrug and four baseline models and complete our project's purpose of verifying SafeDrug's performance against the baseline models.

5.1 What was easy

The replication process for SafeDrug and the baseline models was easy, as SafeDrug and many of the baseline models are publicly available on GitHub. Given that the objective of this project is to verify the performance of SafeDrug against the baseline models, we were able to directly use the published codes and obtain performance results without significant edits or alterations to the source codes. It was also easy to obtain credible results for the comparison between SafeDrug and baseline models. Many of the baseline models have different implementations available for reference on GitHub, it was possible to use baseline model source codes other than those provided in the original paper's GitHub.

5.2 What was difficult

To successfully run and launch some of the replicated baseline models was difficult. Many of the publicly available models include outdated syntax or package references that required updates. Some of the models also had version incompatibilities between different packages, for example, the version compatibility between Python, Theano, and Keras in a version of RETAIN source code. To successfully launch some of these models for a comprehensive comparison, a substantial amount of time was spent on finding the correct and compatible package versions.

5.3 Recommendations for reproducibility

Overall this paper is very easy and straightforward to reproduce. A very minor request would be for the the author to share the final SafeDrug model in the GitHub repository. Due to the randomness in shuffling and run environments, it is very difficult to get the same parameters even with the same code. Therefore, having the original final model would give us a higher level of confidence in the reproduced results.

6 Communication with original authors

The code repository of the original paper is comprehensive and with clear instructions on where to collect the data and how to run the code. Therefore,

we did not contact the original authors at the replication stage. After finishing the replication work, we sent a copy of our final report to the author for feedback and recommendations.

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.

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. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1).

Hung Le, Truyen Tran, and Svetha Venkatesh. 2018. Dual memory neural computer for asynchronous two-view sequential learning.

Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2018. GAMENet: Graph Augmented MEmory Networks for Recommending Medication Combination. *arXiv preprint arXiv:1809.01852*.

Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. 2021. Safedrug: Dual molecular graph encoders for safe drug recommendations. *ArXiv*, abs/2105.02711.

Yutao Zhang, Robert Chen, Jie Tang, Walter F. Stewart, and Jimeng Sun. 2017. Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.