Reproducibility Project Instructions for CS598 DL4H in Spring 2022

Nadia Wood and Diana Gonzalez Santillan
{nadiaw2, dianag4}@illinois.edu

Group ID: 73, Paper ID: 159, Difficulty: Easy
Presentation link (TODO in final paper): https://www.youtube.com
Code link: https://www.github.com/nadiawoodninja/CS598DLHFinalProject

1 Introduction

The paper that we chose to reproduce is titled "INPREM: An Interpretable and Trustworthy Predictive Model for Healthcare", it was written by Zhang, Xianli, et al. and was published in the Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in 2020 (Zhang et al., 2020).

In this paper, a unique predictive deep learning model called INPREM was proposed. This model was trained on the MIMIC-III dataset, as well as a real-world longitudinal EHR database with three cohorts (Heart Failure, Diabetes, and Chronic Kidney Disease). The goal of INPREM was to enhance the interpretability and trustworthiness of clinical predictive models, while still maintaining an optimal performance.

On the one hand, to improve interpretability, a contribution matrix, which helps explain where predictions come from, is generated by a model built with both linear and non-linear parts. The non-linear part contains several different attention mechanisms. On the other hand, to improve trustworthiness, a Bayesian Neural Network is implemented for the diagnosis prediction use case. Using Monte Carlo sampling, the model is able to capture the uncertainty for each prediction generated.

By enhancing interpretability and trustworthiness, the authors aim to make deep learning results more predictable, accessible and useful for healthcare professionals, and in this way they hope to help improve the process of real-life clinical decision making.

This paper's findings are important because they prioritize the usefulness of deep learning models for the actual humans using them. In addition, the authors claim that INPREM performs better compared to all other state of the art machine learning approaches in terms of accuracy.

2 Scope of reproducibility

For this reproduction study we have chosen the claims from the original paper about enhanced interpretability and about improved accuracy (Zhang et al., 2020). The original paper also makes a claim about the trustworthiness of the system based on the uncertainty captured for each prediction using Monte Carlo sampling, but have decided not to include this claim in our reproduction for the sake of simplicity.

2.1 Addressed claims from the original paper

The specific claims from (Zhang et al., 2020) that we are testing in this project are the following:

- The contribution matrix generated by INPREM improves the model's interpretability by showing which medical codes contribute the most for each of the model's predictions.
- The best case accuracy for INPREM outperforms other deep learning models, and its values range from 0.69 for 5 epochs to 0.83 for 30 epochs.

3 Methodology

For this project we were lucky enough to obtain a version of the code titled MINI-INPREM from the authors of the original paper. Therefore we decided to use that code as a starting point for our project. In addition, we were able to address the reproducibility appendix in the original paper (Zhang et al., 2020) to get specifics on datasets, baselines, and implementation details in order to better setup our reproduction.

3.1 Model descriptions

In the original paper (Zhang et al., 2020), there were three main models used to fulfill the claims of the work. In our reproduction we used same

models with the same setup and parameters. The notation used to describe each model can be found in the following table:

C	set of all unique medical codes.
$\|C\|$	number of unique medical codes.
T	total number of visits associated to a patient.
v_i	a specific visit, binary vector of size $\|C\|$.
$v[j]$	the j-th medical code associated to visit v.
X	input matrix of visits.
g	dimension of embedding space for visit representations, set to 256 in paper.
$\odot$	element-wise multiplication of two vectors.
l	number of classes to predict (defined depending on task).
d	dimension of a matrix, set to 265 for all input matrices in paper.
m	number of heads for multi-head attention.

The three original models are described in the following subsections.

3.1.1 Linear Model for Interpretability:

According to (Zhang et al., 2020), the objective for this model is to obtain the embedding $E_v \in \mathbb{R}^{g \times T}$ for each medical visit v in the dataset by modelling the relationships between the medical codes within each visit. Namely:

$$E_v = W_v X \tag{1}$$

where $W_v \in \mathbb{R}^{g \times |C|}$ are the parameters to be learned. An extra embedding layer is also included to encode the order information for each visit after attention is applied. Namely:

$$E_o = W_o O \tag{2}$$

where $W_o \in \mathbb{R}^{g \times 1}$ are the parameters to be learned and $O \in \mathbb{N}^{1 \times T}$ is the order of each visit in time.

After E_v and E_o are obtained, a patient-level representation $E_R \in \mathbb{R}^{1 \times g}$ is obtained by defining a linear mapping such that:

$$E_R = \alpha(\beta \odot (E_v + E_o))^\top \tag{3}$$

where $\alpha \in \mathbb{R}^{1 \times T}$ encodes the non-linear relationship between the patient's visits, and $\beta \in \mathbb{R}^{g \times T}$ encodes the non-linear relations between medical events within each visit.

This representation can then be used to develop a predictive model described as:

$$\tilde{y} = W_c^\top E_R^\top + b_c \tag{4}$$

where $W_c \in \mathbb{R}^{g \times l}$ and $b_c \in \mathbb{R}^{l \times 1}$ are the parameters to be learned.

Softmax is then used to estimate the probability y^* for each prediction:

$$y^* = \text{Softmax}(\tilde{y}) \tag{5}$$

Interpretability for Prediction: As explained in (Zhang et al., 2020), due to the linearity of the model, the contribution for each medical event can next be found by inferring from the predicted $\tilde{y}$ back to the input X. In particular, from equations 1 through 3, we can get that:

$$E_R^\top = \sum_{i=1}^{T} \sum_{j=1}^{|C|} \alpha[i]\beta[:,i] \odot (v_i[j] W_v[:,j] + i W_o) \tag{6}$$

Futhermore, considering equations 4 and 6, we can calculate the contribution of each medical event with:

$$\text{CM}[i,j] = W_c^\top (\alpha[i]\beta[:,i] \odot W_v[:,j]) \tag{7}$$

where $\text{CM} \in \mathbb{R}^{T \times |C| \times l}$ is the Contribution Matrix and $\text{CM}[i,j][k]$ denotes the contribution of the j-th medical event in the i-th visit to the prediction when the predicted class is k (Zhang et al., 2020).

3.1.2 Non-linear Model for Dependencies

As stated by (Zhang et al., 2020), the linear model above is not able to fully capture dependencies between and within each medical visit, even though this information highly contributes to the predictions generated.

To further increase interpretability, the non-linearity associated to these dependencies can be encoded into α and β from equation 3 by first introducing a stacked multi-head attention module, which outputs a hidden state with strong representation power, and then utilizing a sparse attention module, and a variable attention module, both of which respectively weight the importance of different visits, and the importance of different medical codes within each visit (Zhang et al., 2020).

The details on each attention module used are as follows:

Stacked multi-head attention: Formed by multiple layers running in parallel. Each layer gets as input a set of key-query pairs, as well as corresponding values. The key-query pairs are used to get inner dependency weights, which are then used

to obtain new values. Formally (Zhang et al., 2020) define each individual self-attention layer to be:

$$\text{Att}(Q, K, V) = V(\text{Softmax}(\frac{Q^\top K}{\sqrt{d_k}})) \quad (8)$$

where

$$Q = W_1(E_v + E_o) \in \mathbb{R}^{d_k \times T}$$

$$K = W_2(E_v + E_o) \in \mathbb{R}^{d_k \times T}$$

$$V = W_3(E_v + E_o) \in \mathbb{R}^{d_v \times T}$$

and $W_1, W_2 \in \mathbb{R}^{d_k \times g}$ and $W_3, \in \mathbb{R}^{d_v \times g}$ are the weights to be learned by the model.

Moreover, the multi-head attention is obtained by concatenating multiple individual attention layers and fusing everything with a fully-connected layer. Namely:

$$\text{MultiHeadAtt}(Q, K, V) = W_o\text{Concat}(\text{Att}_1, \text{Att}_2, ..., \text{Att}_m) \quad (9)$$

where $W_o \in \mathbb{R}^{d_{model} \times m d_v}$ is the parameter to be learned. The multi-head attention module is then stacked S times to strengthen the semantics of the model. Each multi-head attention is also followed by a feed-forward layer (two 1D convolutional layers, plus ReLU activation) (Zhang et al., 2020).

Sparse attention (visit level): With this module, (Zhang et al., 2020) aims to emphasize visits that contain important features for a diagnosis by augmenting the Softmax(.) in the prediction model with an additional Sparsemax(.) component. This is achieved by setting the visit attention weight, α, from equation 3 to be:

$$\alpha = (\text{Sparsemax}(\delta) + \text{Softmax}(\delta))/2 \quad (10)$$

where $\delta = W_\delta H + b_\delta \in \mathbb{R}^{1 \times T}$ is the correlation vector from the hidden state H, and $W_\delta \in \mathbb{R}^{1 \times d_{model}}$ and $b_\delta \in \mathbb{R}^{1 \times T}$ are parameters to be learned.

Variable attention (medical event level): With this module, (Zhang et al., 2020) aims to emphasize important features (medical codes) within a single visit by setting the medical code attention weight, β, from equation 3 to be:

$$\beta = \tanh(W_\beta H + b_\beta) \quad (11)$$

where $W_\beta \in \mathbb{R}^{g \times d_{model}}$ and $b_\beta \in \mathbb{R}^{g \times 1}$ are the parameters to be learned.

3.1.3 Bayesian Neural Network (BNN) Model for Trustworthiness:

According to (Zhang et al., 2020), BNNs implement Bayesian probability theory with Neural Networks in order to place a prior distribution $p(f)$ over the space of a function f, and then search for the posterior distribution $p(f|D_X, D_Y)$ given the dataset (D_X, D_Y).

In practice, a series of approximations and assumptions need to be made to be able make this posterior distribution tractable using Neural Networks. In (Zhang et al., 2020), an objective function of variational inference is defined, and after the aforementioned assumptions and approximations it can be computed based on the model's predictions.

This objective function can also be rewritten to formulate either the model's Risk Prediction, $\mathcal{L}_{risk}$, or its Diagnosis Prediction, $\mathcal{L}_{esm}$ and this way obtain the uncertainty for each of the model's predictions (Zhang et al., 2020).

3.2 Data descriptions

For this project, as in the original paper (Zhang et al., 2020), we have used the MIMIC-III public dataset. However, the original paper also used some private real-world datasets for Heart Failure, Kidney Disease, and Diabetes that were not available to us due to legal reasons. Because of this and also for the sake of simplicity, we have decided to exclude the Kidney Disease and Diabetes datasets from our reproduction attempts, and focus on the Heart Failure data only.

We were able to find Heart Failure data in MIMIC-III. This is not the same as the data used by the original authors, but after some wrangling it works fine with the model and is able to give us decent results.

In particular, these are the datasets we have used for our reproduction project:

- The publicly accessible MIMIC-III (Johnson et al., 2016) dataset for ICD9 medical codes, found at datasets/diagnosisCode.csv in our repository. This dataset has 1761 datapoints, each corresponding to one meidcal code and containing five features: Row ID, Subject ID, HADM ID, Sequence Number, and ICD9 Code.

- The publicly accessible MIMIC-III (Johnson et al., 2016) dataset for heart failure diagnosis, found at datasets/HeartFinalDataset.csv in

our repository. This dataset 479 datapoints, each corresponding to a medical visit and containing four features: Subject ID, Chart Date, CPT CD, and Has Diagnosis.

Both of these datasets were split into training, validation, and testing sets with a ratio of 75:10:15 which were the same ratios used in the original paper (Zhang et al., 2020).

Note that more details on how to query MIMIC-III, and cleanup the data to get each of these datasets, as well as detailed dataset statistics, can be found in the README file of our git repository.

3.3 Hyperparameters

The code allows us to choose various options for running the INPREM model. A full list of the options available for executing INPREM can be found in the README of our git repository.

We decided to use the default values for most of the options when running the model, as the majority of them matched the hyperparameters stated as optimal in the original paper (Zhang et al., 2020). For example, we left the learning rate at its default of 5e-4, the dropout rate as 0.5, and batch size of 32, among others.

In addition, when executing our code, we explicitly specify the embedding dimension and inner dimensions as 256 (when the default is 128), as this was the value used in the original paper (Zhang et al., 2020).

3.4 Implementation

We used the existing MINI-INPREM code emailed to us by (Zhang et al., 2020) as a starting point. We put a copy of the code in our own repository (link provided at the top of this report) and did the following changes:

- Added our demo data to the folder called datasets.
- Updated the libraries to use Python 3 and installed dependencies
- Modified data import, training, validation, and testing parts to work with our sample datasets.

More details on how exactly we managed to implement the code provided to us by the authors can be found in the README of our git repository, as well as in comments within our main.py file.

3.5 Computational requirements

In the original paper, all experiments were implemented with PyTorch 1.0 on two Nvidia Titan XP GPUs (Zhang et al., 2020). For our reproduction, we have used the "MINI" version if the same code, also written in PyTorch.

In our project proposal, we believed we might need to find a way to access Nvidia Titan XP GPUs for this work (possibly through Google Cloud). Nevertheless, since the version of the code we got is a reduced "MINI" version of INPREM, and because our sample dataset is also smaller than the original, we were actually able train and test our model efficiently in our local machines.

The model took about 50 seconds to be fully trained on 25 epochs and tested, which means about 2 seconds runtime per epoch on average. We ran the code on on a 4-core 2.8 GHz Intel Core i7 with 6 GB 1600 MHz DDR3 memory. For future implementations we might try to run this code with a larger dataset, and more specialized GPUs might be needed at that point.

4 Results

After implementing the data import, training, validation, and testing parts of our main.py file in our MINI-INPREM copy, we were able to run our model for several different epoch counts and calculate the varying accuracy and F1 score.

Although our datasets were not the same as the original paper due to privacy reasons, we tried to keep as many elements as possible the same as in (Zhang et al., 2020). We used the same hyperparameters as the ones described in the paper to get our own sets of results that can be compared to the original ones.

4.1 Result 1: About Interpretability Claim

The original claim was that the contribution matrix generated by INPREM improves the model's interpretability by showing which medical codes contribute the most for each of the model's predictions (Zhang et al., 2020).

Due to lack of access to model logic and appropriate data set, we were unable to replicate the results presented in the paper. The "MINI-INPREM" version of the code provided to us by the authors focused mainly on the prediction part of the model. It did not contain logic to generate a contribution matrix, In order to generate the contribution matrix, EHR data set was needed which we did not

Number of epochs:	5	10	15	20	25	30
Baselines Best Case:	0.6399	0.5840	0.6318	0.7018	0.7687	0.8212
INPREM Best Case:	0.6902	0.6253	0.6626	0.7306	0.7890	0.8314
Our Model:	0.4286	0.8900	0.7143	0.5714	0.5714	0.5714

Table 1: Comparison of ACCURACY results for section 4.2

Epochs	F1 score for our model
5	0.33
10	0.89
15	0.75
20	0.73
25	0.67
30	0.40

Table 2: Additional metric: F1 SCORE for our model

have access to. Hence, it was not feasible for us to reproduce the claim.

In addition, even with a contribution matrix, all we could have done was compare it to that of the original paper to see if there were any differences, but without professional medical knowledge we wouldn't be able to empirically evaluate if our results were better, worse, or similar to those of the original paper.

4.2 Result 2: About Accuracy Claim

The original claim was that the best case accuracy for INPREM outperforms other deep learning models, and its values range from 0.69 for 5 epochs to 0.83 for 30 epochs (Zhang et al., 2020).

Our hypothesis is that our reproduction will have similar performance, as we have used the same hyperparameters and number of epochs to run our experiments. Nevertheless, our results might be dragged away from those of the original paper because our dataset is smaller, and we run the risk of overfitting.

We compare our accuracy results (which range from 0.43 for 5 epochs to 0.71 for 15 epochs) with the original paper's baseline and INPREM best case accuracy results in Table 1. We also calculated the additional metric of F1 scores, which can be reviewed in Table 2. These results are further discussed in section 5.

4.3 Additional results not present in the original paper

We did not have any additional results besides the ones outlined in our previous section. We have been able to run the code provided to us by the original authors successfully by using the same hyperparameters as those outlined in the reproducibility appendix of (Zhang et al., 2020), but we did not have time to try any additional experiments.

5 Discussion

Our reproduction is different from the original paper (Zhang et al., 2020) in many aspects. To begin, we used the "MINI" version of the model that the authors provided to us, as opposed to the full version of the model. In addition, our data sets were smaller and more limited than the data sets used by the original paper. The original data sets were not available to us due to privacy reasons.

The "MINI" version of the code, as well as the reproducibility appendix in the original paper (Zhang et al., 2020) mainly focused on the prediction model and its accuracy results. Due to this, is was hard for us to reproduce the most novel and interesting results of the original paper, namely those concerning trustworthiness and interpretability.

As mentioned in section 4.1, it was infeasible for us to reproduce the contribution matrix mentioned in (Zhang et al., 2020), which is the main feature for the model's interpretability component. Similarly, due to time constraints and the complexity involved, we were unable to generate uncertainties for each prediction, which is the main feature for the model's trustworthiness component.

Despite missing the contribution matrix and uncertainties, we were still able to replicate INPREM's prediction model using the author's "MINI-INPREM" code as a base. The model's

predictions themselves are still a significant result in the original paper due to their increased accuracy compared to other state of the art deep learning methods used as baselines (Zhang et al., 2020).

In section 4.2 we show our results from running the model that we replicated. Although our accuracy numbers do not match those of the original paper (Zhang et al., 2020), we believe that we actually fulfilled our hypothesis. Even though our accuracy is not optimal our F1 scores seem more promising and closer to the performance of the original INPREM.

We believe our results in section 4.2 are caused by our lack of suitable data, and not due to our code's inadequacy. Our model behaves similarly enough to the original one, but naturally shows some overfitting due to the small size of our dataset. This is why F1 scores, which are more sentitive to imbalanced or small datasets, show a better performance.

5.1 What was easy

It was easy for us to get in contact with the author and get a version of INPREM's code from them, it was also relatively easy for us to understand their code and find where we needed to add additional lines in order to import our data, train, validate, and test the model. Moreover, it was not complicated for us to get the MIMIC-III demo data (Johnson et al., 2016), and after some inspection we were also able to find suitable heart disease data that could replace the private datasets we were missing.

In general we think the original paper (Zhang et al., 2020) is very well written and does a good job at transmitting why their work is novel an important, so it was easy for us to understand the relevance of the original paper's results, as well as the model components and settings. This made it easy for us to summarize the model in section 3 and replicate the model to the best of our ability in our code.

5.2 What was difficult

Overall everything to do with data and data access was harder than we would have liked. Due to the nature of healthcare data in general, it was difficult for us to find a suitable dataset that we could actually access. Once we found accessible data, we needed to clean and wrangle this data in order to make it fit to what the "MINI-INPREM" model was expecting as input, which was also tedious.

We found it difficult to replicate results beyond the prediction model itself, as it was unclear from the reproducibility appendix of (Zhang et al., 2020) how exactly the contribution matrix and prediction uncertainties are generated and used. In addition, we did not have contact with any medical professionals who could help us empirically judge our results.

5.3 Recommendations for reproducibility

The paper itself does a very good job at outlining the model components and hyperparameters. Nevertheless, it does not get into detail when explaining how to get the interpretability and trustworthiness results which are in our opinion the most interesting part of the paper. It would be great for future reproducibility if more details could be included in that front.

6 Communication with original authors

When we chose this paper as one of our potential papers in the project proposal, we emailed all of the authors listed for the paper to see if they could share with us any code, data, or useful tips.

Xianli Zhang replied to us promptly and was very helpful in providing us with the "MINI" version of INPREM's code for us to be able to reproduce their work for our class. When the attachment they sent to us got removed by the Illinois mailing system, they even took the time to put the code in a Google Drive for us to access.

While communicating with Xianli Zhang, we also asked if it was possible for them to share the non-MIMIC-III datasets they used disease detection. However, they replied that they were unable to help us in that front since the datasets they used were private and it would be illegal for them to share them with us. They also refused to give any information about the schema of the datasets used.

Overall, we consider that the communication with the author was good and successful, we understand the legal and privacy concerns they had and also understand that due to their busy schedule they were not able to help us further in the code implementation. We have included the full reproduction of our email communications in the "Communication With Authors" section of our git repository's README (link to repo can be found at the top of this paper).

References

A Johnson, T Pollard, and R Mark. 2016. Mimic-iii clinical database (version 1.4). *PhysioNet*.

Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. 2020. Inprem: An interpretable and trustworthy predictive model for healthcare. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 450–460.