

Reproducibility Project of "Identify Susceptible Locations in Medical Records via Adversarial Attacks on Deep Predictive Models" for CS598 DL4H in Spring 2022

Nivedita Chatterjee and Nissan Azizov
{nc19, nazizov2}@illinois.edu

Group ID: 155, Paper ID: 151(Easy)

Presentation link: <https://uofi.box.com/s/8v5frko6xzskhmhhw70eag63aed67qk4>

Code link: https://github.com/na21/dlh_project

1 Introduction

The general problem the work is trying to do is to address the vulnerabilities in deep learning models for EHR (Electronic Health Record) data. They claim that existing models may demonstrate excellent performance but can be extremely sensitive to inputs that may have some small negligible changes. They refer to them as "adversarial examples" where a slight alteration to a patient's medical record may alter the results of the deep learning model. The paper proposes a framework that learns about attacks that target the RNN/LSTM models that have EHR inputs that can leverage it to screen medical records of patients. This framework can create a sensitivity or susceptibility score to help medical professionals pinpoint entries that can cause high damage if not accurately recorded or measured.

2 Scope of reproducibility

This paper proposes a framework consists of "LSTM predictive Model" and "adversarial attack procedure" to identify the susceptible location in EHR. Here authors introduced a measure call "susceptibility score" at patient level and population level which will help Clinician to identify the location of the record, in this case the vital which can cause high damage if not recorded correctly or tampered.

2.1 Addressed claims from the original paper

We tested below claim:

- Performance of 5-fold cross validation for LSTM.
- Maximum perturbation and when it is changing while adversarial generation

3 Methodology

For this paper authors used MIMIC3 dataset - mainly chartevent,labevent time series data. We are reusing Author's original code for predictive model and Adversarial generation. We connected with Author(Fengyi (Andy) Tang) on LinkedIn for guidance about the datasets used for the main code. He suggested to look up his "Urgent-care" code pre-processing step which can be used for Med-attack after few changes. Details are described in the implementation section Code: [https://github.com/illidanlab/med-attack\[1\]](https://github.com/illidanlab/med-attack[1]) We could not produced "susceptibility score" which is main output of the framework to determine susceptible location in a EHR at paptient and population level. Basic methodology is very well demonstrated in the paper using below diagram Medical record (patient record with 19 feature -time series data) – RNN network - Predictive model (LSTM) – Adversarial generation (iterative optimization) – Adversarial Medical record – susceptibility score.

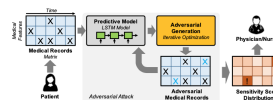


Figure 1: Methodology

3.1 Model descriptions

This dataset is highly imbalanced with respect to mortality outcome(only 11 percent is deceased). To handle this scenario authors down-sample observations from the negative class during train-test splitting using StratifiedKFold 5 fold cross validation is used for train and test data. model is trained with 19 feature from dataset. 27616 multivariate time series with 19 variables across 48 time stamps. this is the final dataset used for this model

3.1.1 Model Architecture

RNN network contains 3 layer . 1 LSTM layer(128 hidden node) 1 fully connected layer(32 hidden node) softmax layer(2 hidden node). for the loss function "softmax_cross_entropy_with_logits" is used here. AdamOptimizer is used for optimizer. Gradientdescent optimizer is used with learning rate -0.02 is user for generating Adversarial Examples

3.2 Data descriptions

We have used MIMIC3 dataset. This dataset contains health-related information for over 45,000 de-identified patients who stayed in the critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. MIMIC-III contains information about patient demographics, hourly vital sign measurements, laboratory test results, procedures. This experiment uses records from a collection of patients, each being a multivariate time series consisting of 19 variables from vital sign measurements (6) and lab events (13). Vital signs include heart rate (HR), systolic blood pressure (SBP), diastolic blood pressure (DBP), temperature (TEMP), respiratory rate (RR), and oxygen saturation (SPO2). Lab measurements include: Lactate, partial pressure of carbon dioxide (PaCO2), PH, Albumin (Alb), HCO3, calcium (Ca), creatinine (Cre), glucose (Glc), magnesium (Mg), potassium (K), sodium (Na), blood urea nitrogen (BUN), and Platelet count. These details can be obtained from Chartevents and labevents datasets with "DItems" and "Dlabitems". icustays, admissions tables will be used to determine cohort for this experiment

- Defficulty with Handling dataset : Lot of missing value and outlier.
- Data is not balanced between alive and dead

3.3 Hyperparameters

- We tested changing using learning rate and epoch change combination measured the performance based on AUC,f1 score metric
- We have also added dropout layer in the model and recorded performance
- We changed batch size and recorded performance

3.4 Implementation

- **Preprocessing** - We have tried many different ways to prepare data sets for the main code. After we connected with Author we followed preprocessing steps as suggested by the paper on MIMIC3 data set[3] and used some part of the code from urgent-care-comparative[5] repository. For this step
 - we initially tried handling the data and create the dataset using code from scratch but we faced a lot of challenges handling and processing data and ended up with wrong format.
 - Next we tried to use already available MIMIC3 bench mark code[7] to generate benchmark dataset .After running that while exploring the output we saw the dataset was not inline with our code input requirement
 - Finally we used urgent-care-comparative[5] repository preprocessing code some part of main with our changes which generated correct input dataset for main prediction model code.

[illegible]

Figure 2: Feature data

```
[(Pdb) Y_tr
array([[0., 1.],
       [0., 1.],
       [0., 1.],
       ...,
       [0., 1.],
       [0., 1.],
       [0., 1.]])
```

Figure 3: Label data

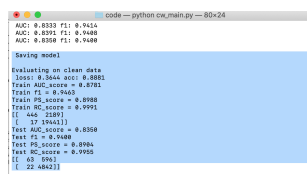
- **Final steps to produce input dataset**
 - – We downloaded MIMIC3[2] data fromPhysionet

- Using postgres build scripts from MIMIC-Code[6] git repository loaded the data in local postgres Database
- We had to make several changes to the pivot scripts from MIMIC-Code[6] repository to create pivot tables and then exported that data to .csv files
- Above step created .csv files were used as input to the urgent-care-comparative[5] preprocess.py scripts. This will generate few .mpy and .pkl file
- As per authors suggestion we created prefiles.py file which creates final input .pkl files in 5 sets for main program

Source code for 5 fold predictive model and Adversarial generation:

- We have downloaded this code from med-attack[1] set up the local python3.5.6 environment using anaconda. This code needs tensorflow 1.1+
- after that we ran cwmain.py [1] this code first trains the model and then creates adversarial data and runs prediction until the prediction value changes.
- This process creates one Advmetric.pkl file as output with 2 main metric "maximum perturbation" "average perturbation"

We successfully created the exact input data needed for this experiment. We are able to generate adversarial data and related metric. We tried to create



```

code - python cwmain.py - B0x24
AUC: 0.8533 F1: 0.9434
AUC: 0.8392 F1: 0.9488
AUC: 0.8358 F1: 0.9488

Saving model
Evaluating on clean data
Jacc: 0.5046 acc: 0.8882
train AUC_score = 0.8781
train F1 = 0.9465
train PR_score = 0.8988
train RC_score = 0.9993
[[ 445 2287]
 [ 17 15451]]
Test AUC_score = 0.8359
Test F1 = 0.9488
Test PR_score = 0.8984
Test RC_score = 0.9955
[[ 63 3961]
 [ 22 48422]]

```

Figure 4: Prediction Model Evaluation

"susceptibility score" which is described in paper. We understood the process and how to produce it but We realized that there is not enough details about input metrics(global maximum perturbation (GMP), global average perturbation (GAP) and global perturbation probability (GPP)) and how to calculate them to produce "susceptibility score". We were in contact with Author Andy who informed this part was written by Mengying Sun.

We tried contacting the author using email and Linkedin but didn't get any response yet. This is why we are unable to produce susceptibility score.

3.5 Computational requirements

We are running this code in our local system GTX 1070 GPU and MAC laptop with CUDA support In this set up the cwmain.py[1] code to generate initial auc,f1,recall,precision score and adversarial metrics takes more than 40 hours. This makes running this code with hyper parameter very challenging. We tried Saving the model and running with small adversarial generation set. actual set up is to run for around 4932 iterations each fold . When we reduced to pick only 10 ,it took 3:02 mins so fold 1 will take 25.35 hours.

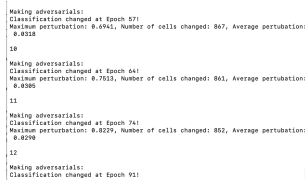
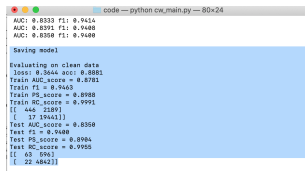
GPU Configuration: We found that the GPU was not being used when running cw_main[1] so we attempted to rectify this for improved performance. We thought that utilizing the GPU could greatly increase the speed over the 4 to 5 days expected runtime. We installed CUDA drivers, got the GPU working in WSL2 and it finally showed up with "nvidia-smi". However, the code was written in Tensorflow version 1.15 or lower and the GPU we have (3070TI) was not working in WSL2 with Tensorflow version 1, only with version 2+. We then attempted to convert the code to support Tensorflow 2+. After much debugging, we found a way to get the GPU working in WSL2 in Tensorflow 1.15 by following instructions from Microsoft here: <https://docs.microsoft.com/en-us/windows/ai/directml/gpu-tensorflow-wsl>.

We fixed it by installing Python 3.5, 3.6 or 3.7, Tensorflow 1.15 and tensorflowdirectml in our conda environment. **The added speed of the GPU greatly increased our speed for model generation from CPU only usage.** We went from 9-10 minutes to create models per fold to 4 minutes with the GPU. However, generating adversarial did not get much increase in performance and our estimations still show that it will take 4-5 days to run that

4 Results

We were able to successfully produce the input dataset. We reproduced and ran prediction model which gave comparable result like than the paper.

We completed data prep section - Imputation, Padding, Normalizing. MIMIC 3 has missing value and imbalanced data. We chose ICUStay patients

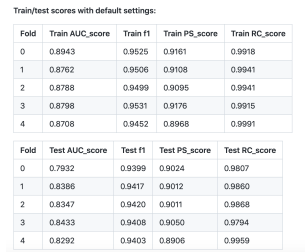


with valid hospital admission id. Base model code and adversarial generation is running without error

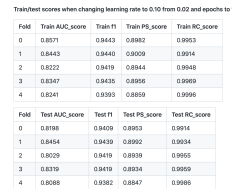
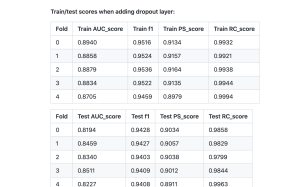
We were able to generate model evaluation metric and adversarial metric. With our input data we achieved better Auc and Precision score.

3	0.8798	0.9531	0.9178	0.9915
4	0.8708	0.9452	0.8968	0.9991

Fold	Test AUC_score	Test f1	Test PS_score	Test RC_score
0	0.7932	0.9399	0.9024	0.9807
1	0.8386	0.9417	0.9012	0.9860



We added dropout layer which degraded the AUC score

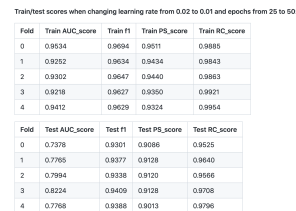


spect to the dataset without dropout layer and less learning rate

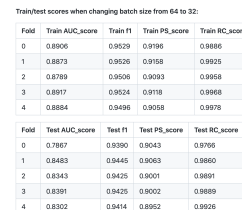
We were also able to produce maximum perturbation and perturbation



7.2 Result5



7.3 Result6



8 Discussion

around the steps and sample code or pseudo code would have been very helpful. We kept our focus on prediction model and adversarial generation. We reproduced and did our testing on these two. As there was no clear direction on cohort selection for this paper. We have used icustay patient with valid(not null) hospital admission id. This gave us better f1 score than the original paper. We concluded model without dropout layer gave much better result in this case

8.1 What was easy

Running the author's code for model train and prediction was easy. We were also able to run the "adversarial generation" part of the code. This code ran successfully and generated corresponding metric

8.2 What was difficult

Preprocessing steps mentioned in the paper was very difficult. We spent most part of our effort in this section. We started communicating with the author to get help with this step. We faced below challenges.

- We could not use any existing benchmark data because it was matching the required format and pattern author used in the paper.
- Most of the steps and instructions mentioned in the "urgent-care"[5] repository were not working due to version upgrade of postgres. This repository was suggested by author to check for preprocessing script
- We had to make several changes to the MIMIC-code[7] repo sql scripts to generate the input data for preprocess scripts
- Many instructions were not clear about creating base datasets to run preprocessing script which author suggested.
- long run time of main scripts made it very challenging to alter and test the parameters
- We used subset of data for adversarial generation because with whole dataset model is estimated to run 4 to 5 days in macbook . We ran for around 50 hours with bigger dataset but it didn't complete
- For analyzing Max perturbation and average perturbation extra information and code is needed from author Mengying Sun who was not reachable using email and linkedin
- "Susceptibility score" generation does not have clear instruction and enough details to reproduce We spent time trying to understand reproduce this piece . We also tried reach out to the author Mengying Sun for this piece. We didn't receive any response yet

8.3 Recommendations for reproducibility

- Refer to urgent-care code [5]. Some parts of the code can be used with modification for preprocessing and data preparation
- MIMIC-code[6] postgres pivot sql code does not run on newer version of postgres. It needs several changes
- Author Fengyi (Andy) Tang is very responsive and helpful. Contacting in linkedin was quicker
- Setup local environment properly as per requirement. python3.5.6 works very well
- Keep enough time to run the main code when running local
- For "Susceptibility score" piece please try to contact Mengying Sun, paper does not have enough details.
- It will be very helpful if authors can add more details related codes around susceptibility score, preprocessing in the med-attack[1] repository

9 Communication with original authors

We have sent email to one of the author Fengyi (Andy) Tang to get more details about the dataprep and final datasets they have prepared for the experiment. Finally we connected with him in linkedin and started getting guidance from him.

We received initial information about urgent-care [5] repository which helped us a lot to get through the preprocessing and produce input data set for main code

There were few differences between label dataset which we needed and the one produced by preprocessing step. He helped us to understand how to proceed there. We prepared our final data processing step based on this input

He suggested to reach out to Mengying Sun for susceptibility score code details but we didn't receive any response from her

References

- [1] <https://github.com/illidanlab/med-attack/tree/master/code>
- [2] AlistairEWJohnson, TomJPollard, LuShen, HLehmanLiwei, MenglingFeng, 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 (2016), 160035
- [3] https://people.cs.pitt.edu/~jlee/note/intro_to_mimic_db.pdf
- [4] HrayrHarutyunyan, HrantKhachatrian, David-CKale, and AramGalstyan. 2017. Multitask Learning and Benchmarking with Clinical Time Series Data. *arXiv preprint arXiv:1703.07771* (2017)
- [5] <https://github.com/illidanlab/urgent-care-comparative>
- [6] <https://github.com/MIT-LCP/mimic-code/blob/main/mimic-iii/>
- [7] <https://github.com/YerevaNN/mimic3-benchmarks>