Deep-insight visible neural network (DI-VNN) for improving interpretability of a non-image deep learning model by data-driven ontology

# Introduction

In this Supplementary Information, we describe details on this study following chronological order of the deep-insight visible neural network (DI-VNN) pipeline. We would use prelabor rupture of membranes (PROM) as an example. There are three of six sections corresponding to some sections in the main text, which are respectively Introduction, Software and equipment, and Procedure. Along with this PDF document, we also provide R Markdown (.Rmd) containing the same texts with this document but including the programming codes for the data analysis in-between of these texts. The R Markdown are available in <https://github.com/herdiantrisufriyana/divnn_proc>. To get raw data, one need to request an access from the BPJS Kesehatan for their sample dataset published in August 2019. Up to this date, there are three sample datasets they published in February 2019, August 2019, and December 2020. For the first and second versions, a request is applied via <https://e-ppid.bpjs-kesehatan.go.id/>, while the third is applied via <https://data.bpjs-kesehatan.go.id>. To preprocess the raw data into the input dataset of this study, follow the codes of the R Markdown in <https://github.com/herdiantrisufriyana/medhist/tree/main/preprocessing>.

# Software and equipment

We set up a programming environment for this study. Bioconductor was utilized as described in the main text. There were 179 R packages which are 10 base packages, 48 other packages, and 121 dependencies.

# Procedure

## Step 1

The data source was a sample dataset of the whole health insurance database during 2015 and 2016 by cross-sectional design. Stratified random sampling was applied. The strata variable was constructed from 66,072 combinations of all the healthcare facilities (*n*=22,024) and category of family, which were: (1) a family of which members never visit the healthcare facilities; (2) a family of which members have visited only primary care; and (3) a family of which members have visited all levels of care. For each stratum, one to ten families were randomly included. This means only 10 families were randomly included if more than that number, resulting 586,969 families with 1,697,452 subjects.

We conducted non-essential data cleaning, e.g. revising the inconsistent name of states, estimating the healthcare identifiers, *et cetera*. These procedures were parts of our R package of medhist 0.1.0. No sampling was conducted.

After the non-essential data cleaning, we applied retrospective cohort design, as described in the main text. For pregnant women, we use several codes for determining delivery or immediately after delivery care. The 220 codes are described.

We conducted data preprocessing after defining the target population and sampling it retrospectively. Demographics were included as categorical variables for causal factor we used as variable of interests. Then, We computed a number of days for a code, or any code representing a causal factor, in the latest encounter before each visit.

To ensure historical rates defined by derivation set only, we need to conduct data partition before continuing the downstream analysis. Therefore, historical rates were not derived by involving validation set.

Deriving historical rates is relatively time-consuming. If one eventually uses this method for predictive modeling, it is efficient to conduct any filtering of predictors before derivation of historical rates. Therefore, more time can be saved and one may not need to run expensive computation.

All candidate predictors, including non-demographical causal factors, have non-zero variances. There were 460 candidate predictors fulfilling this criterion. We also showed in the same table that there are 426 candidate predictors without perfect separation.

We excluded the diagnosis/procedure codes that may leak the outcome information. We only used the existing codes in the training set to determine outcome-leaker codes based on the previous codes for determining delivery or immediately after delivery care. There were 54 codes that may leak the outcome. All of them were also irredundant.

We also determined causal factors as the candidate predictors. These can be an example how to conduct the data transformation on a variable represented by multiple codes of diagnosis and procedure. We combined these factors with other variables that assign a single code.

We inferred the nationwide historical rates given the day number from a code encounter to current visit for each candidate predictor, as described in the main text. This used irredundant candidate predictors with non-zero variances and no perfect separation in training set only.

The candidate predictors were transformed into the historical rates. We conducted this step in all data partitions within a provider-wise dataframe. But, we used the nationwide historical rates using derivation set only. If one uses this data transformation method for predictive modeling, the derivation set may be the same with a training set.

## Step 2 to 13 and 19

Previous data partition had not held out instances for calibration yet. This took 80% of training set. We also gave different weights for event and nonevent by including censored outcome. Hyperparameter tuning, final training, and calibration were conducted by bootstrapping for 30 times. The same resampling methods were applied for both classification and estimation tasks. Parallel computing by multiple central processing units (CPUs) were applied for calibrating the model.

We used nationwide, pre-calibration training set to conduct feature selection by differential analysis. After quantile-to-quantile normalization, we conducted differential analysis as described in the main text. Only candidate predictors that showed adjusted *p*-values <0.05 were selected for 1-bit stochastic gradient descent transformation.

Demanding different statistical assumption, we used unnormalized candidate predictors for creating a feature map (i.e. ontology array) and a network architecture (i.e. ontology network). Both procedures need a distance/similarity matrix. Standardization was applied by subtracting each value with feature-wise average and dividing it with feature-wise standard deviation. Then, we computed a feature-to-feature Pearson correlation matrix.

For creating a feature map, we projected the filtered candidate predictors onto three dimensions, as described in the main text. Meanwhile, using the same correlation matrix, we applied CliXO algorithm, as described in the main text. Each ontology array was fed to a block of neural network, i.e.  Inception v4-Resnet.

## Step 14 to 18 and 20

To get a representation, the computation is conducted by a series of iterations. A backpropagation algorithm was used each time. The computation was applied by the loss function, as described in the main text.

## Step 21

The DI-VNN calibration may be considered, as described in the main text. The calibration used a ~20% split of a training set. The calibration set may be used to compare the predictive performance with other models, if any.

## Step 22

Population-level data exploration is briefly described in the main text. We can provide interactive figure and table of DI-VNN. We used all pre-calibration set for population-level exploration. For simplicity, we only demonstrated population-level exploration of the classification model.

## Step 23

For individual-level exploration, we prepared an example dataset. Intermediate layers are extracted from this example. For simplicity, we only demonstrated individual-level exploration of the classification model. Description for this exploration is already clearly described in the main text.