Human and machine learning pipelines for responsible clinical prediction using high-dimensional data

# Introduction

In this Supplementary Information, we describe details on this study following chronological order of human and machine learning pipelines for responsible clinical prediction using high-dimensional data. 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/hmlp>. 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

## Programming environment

We set up a programming environment for this study. Bioconductor was utilized as described in the main text. There were 198 R packages which are 9 base packages, 53 other packages, and 136 dependencies.

# Procedure

## Step 1 to 6

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 (Table 7 in Supplemental Spreadsheet).

## Step 7

We conducted data preprocessing after defining the target population and sampling it retrospectively. Demographics were included as categorical variables for causal inference. Then, we applied systematic human learning, as described in the main text, to determine what were causal factors that can be inferred from our dataset. We also computed a number of days for a code in the latest encounter before the time of prediction, including those by codes as a causal factor.

## Preparation of validation sets for Step 27

To ensure all inference or derivation using training set only, we need to conduct data partition before continuing the downstream analysis. Therefore, historical rates, PCs, and modeling were not derived by involving validation set.

## Step 8

We conducted causal inference as described in the main text. This will help us to include only the confirmed causal factors as candidate predictors before conducting pre-selection of those candidates to fulfill quality control of predictors in the main text. We included causal factors of which the data were available in training set. Details on this information and ICD-10 codes or demographical variables for each candidate of causal factors are shown in the next section.

## Step 9 to 12

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.

By systematic human learning and causal inference using available data, we also determined causal factors as the candidate predictors. There were 27 first- and 10 second-level factors of PROM. Only data for 12 out of 27 causal factors were available in training set. Either the diagnosis/procedure codes, or demographical variables (not included as candidate predictors), for causal factors are also described.

## Step 13

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 in all data partitions.

## Step 14

The historical rates of all candidate predictors were fitted to a principal component (PC) model. Only training set was used for the model fitting. We applied 10-fold cross validation to estimate weights for all candidate predictors in each PC.

## Step 15 to 17

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, as described in the main text. For hyperparameter tuning, we applied 5-fold cross validation, instead of 10-fold as applied for PC modeling. Meanwhile, the final training and calibration for each model 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 training all models.

## Step 18 to 21, and 26 to 27

We applied the tuning grids and the training configurations for all models, except DI-VNN which required several modifications. This is already described clearly in the main text. More details will be described for DI-VNN in the next section.

## Step 22 and 26 to 27

We applied different pipelines for DI-VNN compared to other models. Please kindly find description of this model in the protocol [a protocol citation]. We used the same example with this pipeline.

## Step 23 to 25, and 27 to 28

Model evaluation is already clearly described in the main text. We started from evaluating the calibration measures of all models for classification task. Then, we chose the best model for classification by AUROC using only internal validation (calibration split) bootstrapped for 30 times. Using the same subset, the best estimation model was also chosen.

We also confirmed the robustness of the best models using external validation sets. Uncertainty intervals were also computed by bootstrapping for 30 times. A clear description on model validation is already given in the main text.

## Step 29

For web application, we prepared an example dataset, user interface, and processing script at the side of server computer. No line of codes for the web is included in the R Markdown or R Script. Description for this web application is already clearly described in the main text.