STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Predicting Mortality from MIMIC III Using Deep Learning Models

Authors: Haohang Li, Baihao Huang, Yuhui Ren

Instructor: Professor Emily Liu

1. Introduction

Reasonable allocation of medical resources and targeted saving of patients' lives are always big tasks for medical workers. However, with advances of technology, can we employ AI to predict patients’ mortality, so as to save more lives? Based on this motivation, we applied deep learning methods to create several models for patients’ mortality prediction.

The dataset we used is MIMIC III, which contains archived electronic health records (EHRs) collected from patient hospital visits. With selected both structured and unstructured features from MIMIC III, and embedding (both doc2vec and word2vec) for the unstructured data, we developed four deep learning models, namely ANN, fusion CNN+ANN, fusion CNN+ANN+CNN, fusion CNN + ANN+RNN, in this project. The model with fusion CNN+ANN+CNN outperforms, with an accuracy of 83%.

2. Literature Review

Electronic Health Records are a medical system that records patient health conditions, such as vital signs, lab tests (structured data), and clinical notes (unstructured data). Since the adoption of EHRs increased from 9.4% to 83.8% from 2008 to 2015 (Henry et al. n.d.), it raises machine learning researchers’ interest in conducting health care research. Recently, various studies showed that the robust predictive power of the deep learning approach could help improve health care and reduce cost (Zhang et al. 2020).

The critical health care dataset MIMIC III is widely used by many researchers (Ding et al. 2021; Hu et al. 2021). In this dataset, the health information of over 5,000 patients who stayed in the intensive care units is recorded entirely, including the procedure codes, diagnostic codes, imaging reports, hospital length of stay, and clinical text.

Converting textual data to numeric data is one of the challenges in our project. Finding the appropriate embedding method is worth discussing. Many previous works have been done in this field. (De Vine et al. 2014) demonstrate that the skip-gram neutral language model can be adapted to encode the EHRs. Furthermore, his work was expanded by (Choi et al. 2016), they propose that a minor modification can sufficiently increase the embedding performance for EHRs, and they also introduce a measure for evaluating the goodness of embedding. By learning from these researchers' work, we were thinking that if these embedding methods could apply to the unstructured data in MIMIC III, for example to embed patients' procedures as numerical data and then feed them to our models.

With processed embedded data, many researchers use deep learning methods to solve real life problems. (Ding et al. 2021) utilize logistic regression and ANN to predict mortality for people with acute pancreatitis. (Zhang et al. 2020) propose that using fusion CNN model and fusion LSTM model to combine structured and unstructured data can lead to better performance on their three prediction tasks.

And many new deep learning models are being proposed recently, by applying various fusion architectures of neural networks and with help of combined structured and unstructured data to improve model performance.

3. Data

3.1 Dataset Description

MIMIC-III (Medical Information Mart for Intensive Care III) is a large, freely available health-related database containing 26 relational tables, which includes information such as demographics, vital sign measurements, laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality. The dataset supports a diverse range of analytic studies spanning epidemiology, clinical decision-rule improvement, and electronic tool development.

The target of our deep learning models is in-hospital mortality, which we got the information from *ADMISSIONS* table. If a patient has a death time, then we will flag this patient for in-hospital mortality.

The features we used can be divided into five parts:

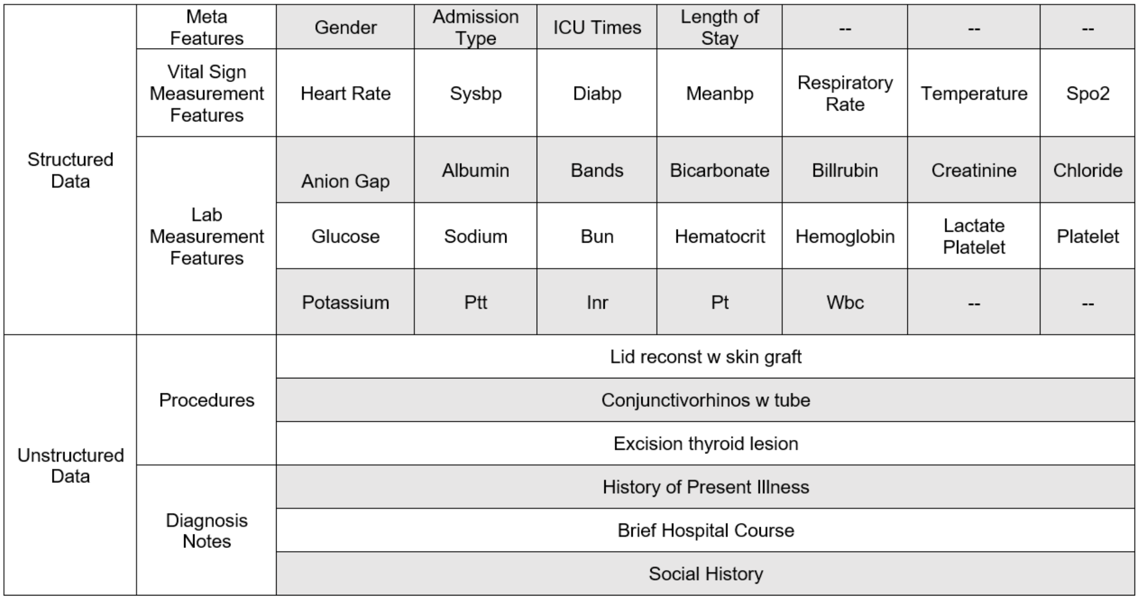
Firstly, general features include *gender*, *length of stay* (how many days a patient stayed in ICU), *ICU_Times* (how many times a patient visits ICU), *admission_type* (how many times a patient was admitted by Elective, Emergency and Urgent units).

Secondly, a variety of features are from *CHARTEVENTS* table, which indicates vital sign observations for patients, e.g., *Heart Rate*, *MeanBP* (mean blood pressure), *Respiratory Rate*, etc. Due to the lack of medical domain knowledge, we choose to adopt the features mentioned in Zhang et al. 2020 as our features.

Thirdly, there are some features from *LABEVENTS* table, which indicate laboratory measurements for patients both within the hospital and in outpatient clinics, e.g., *Hemoglobin*, *Glucose*, *Chloride*, *Bicarbonate*, etc. Since each item could be measured several times for one single patient, we will process the measurement results for each item as one sequence, and each patient has many sequences indicating he or she has many items tested.

Next, we also used some unstructured data from *PROCEDURES_ICD* table, which contains ICD9 (International Classification of Diseases) codes of patients' every visit, indicating what exact procedures are done for the patients. We applied word embedding for these codes, to find out some semantic relations among the codes for patients.

Another part of unstructured data is from *NOTEEVENTS* table, which contains textual information made for patients, such as *History of Present Illness*, *Brief Hospital Course*, and *Social History*. Again, to find out semantic relations in these texts, we applied *doc2vec* method to convert non-numerical features to numerical features.



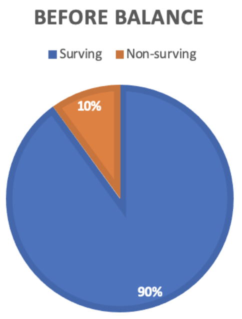
Structured Data	Meta Features	Gender	Admission Type	ICU Times	Length of Stay	--	--	--
	Vital Sign Measurement Features	Heart Rate	Sysbp	Diabp	Meanbp	Respiratory Rate	Temperature	Spo2
	Lab Measurement Features	Anion Gap	Albumin	Bands	Bicarbonate	Billrubin	Creatinine	Chloride
		Glucose	Sodium	Bun	Hematocrit	Hemoglobin	Lactate Platelet	Platelet
		Potassium	Ptt	Inr	Pt	Wbc	--	--
Unstructured Data	Procedures	Lid reconst w skin graft						
		Conjunctivorhinos w tube						
		Excision thyroid lesion						
	Diagnosis Notes	History of Present Illness						
		Brief Hospital Course						
		Social History						

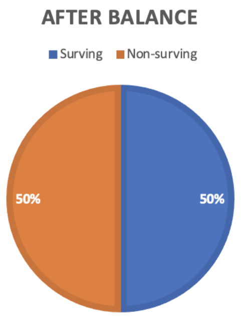
3.2 Dataset Preprocessing

3.2.1 Extract Subset and Under-Sampling

Because of the overwhelming data volume in MIMIC III dataset, it is very convenient to use SQL to select and fetch the data that we need in our project. Totally, there are 10 tables selected in our dataset. However, not every patient appears in each table. In order to fully utilize the data from these 10 tables, we cross-checked each table to find patients who appear in all these 10 tables and extract their information as our features. The SQL queries code are available in the appendix A.

In addition, since the number of surviving patients and the number of non-surviving patients in our data are not balanced, we implemented the under-sampling methodology to balance the two types of patients, so that there are 50% patients survived and 50% patients died in our final dataset.





3.2.2 Feature Preprocessing

3.2.2.1 General Features

Features	Preprocessing Description
Admission	● There are 3 types in this feature, Emergency, Elective and Urgent. ● We will count the times of admission to these three types of ICU for each patient and treat each admission type as a feature.
Gender	● We use one-hot encoding method to denote patients' gender. ● [1, 0] represents the male. ● [0, 1] represents the female.
ICU-Times	● The number of times that the patient discharges from ICU.

3.2.2.2 *LABEVENTS* Processing

Since the *LABEVENTS* and *CHARTEVENTS* are recorded as time series. For different patients, the total number of measurement items and time steps are varying. We will perform the following steps to pre-process *LABEVENTS* dataset and *CHARTEVENTS* dataset:

Step 1: Cleaning abnormal values. For patient measurements, sometimes, the measurements are recorded as strings instead of numeric values (e.g., if measurement INR is less than 1, its value will be recorded as '< 1.0'). Since these observations do not take up much of our entire dataset and our team lacks medical domain knowledge, we will remove them from our *LABEVENTS* dataset and *CHARTEVENTS* dataset.

Step 2: Extract patients. For each patient, we will arrange the time series for different observation items to matrix form while the column direction is time steps and row direction is different measurement items. Here is a simple for a patient table:

Date	Heart rate	Temperature
2020-01-01	89.9	37.2
2020-01-02	90.1	36.8
2020-01-03	87.9	36.7

Step 3: Padding. For each patient, not every lab item will be performed and not every vital signal will be measured. The items that are not recorded will be filled with a zero sequence. Because patients cannot perform multiple items at the same time, there will be some items with blank values for a certain period. We will use nan value to fill that blank. Here is an example:

$$Time\colon \left[t_1, t_2\, , t_3. t_4, t_5\right]$$

$$Item1\colon [nan, nan]\left[I^1_3, I^1_4, I^1_5\right]$$

$$Item2\colon [I^2_1, I^2_2, I^2_3][nan, nan]$$

Step 4: Make up the length. Going back to step 3, we replace the values for lab measurements nan with zero. Since each patient has a different timeline length, we will use zero padding to make the patients have the same timeline length.

$$Time\colon \left[t_1, t_2\, , t_3. t_4, t_5\right]$$

$$patient1\colon \left[I^1_1, I^1_2, I^1_3\right][nan, nan]$$

$$patient2\colon [I^2_1, I^2_2, I^2_3, I^2_4, I^2_5]$$

3.2.2.3 *CHARTEVENTS* processing

Calculate the mean and standard deviation: For each item, we will ignore the abnormal value to calculate its average and standard deviation as our features. Mean can be used to capture the average and standard deviation can be used to detect variation of the observation. For each measurement there is a mean and a variance, and the different means and variances are treated as separate features.

3.3 Embedding

3.3.1 Embedding for patients' medical notes

In our project, we implement Doc2vec to present the medical notes data from *NOTEEVENT*. Due to the large number of medical records in *NOTEENVENT*, we only adopt history of present illness, social history, and brief hospital courses as our new input for our model. History of present illness contains the information of illness that the patients experienced. Social history records the interaction of different groups in society. Brief hospital courses provide more detail about the sequence of events from admission to discharge in a hospital. Based on our domain knowledge, these data have enough information to explain the mortality of a patient.

To embed the patient medical notes into vectors, we firstly use regular expression to extract the wanted data from *NOTEVENT*. Then we use tokenizer to split the data into a single word. After completing this procedure, we send the word sequence to the doc2vec model to generate a 128-dimensional document vector. Since there are three different texts, we will train an individual model for each text. Considering different patients have different number of texts, we will use uniform distributed random number generator to create a padding sequence to make up the missing part so that we can ensure each patient has the same dimensional features.

3.3.2 Embedding for Patients' Procedures

Word2Vec is a model to form or create word embeddings. It's a modern way of representing words , wherein each word is represented by a vector (array of numbers based on embedding size). *Word2Vec* finds semantic or syntactic relations among the words which was not possible by traditional TF-IDF or frequency based approach. When we train the model, each word gets a point in a dimensional space where it learns and groups the words with similar meanings. In this project, we applied *Word2Vec*, specifically Skip Gram) to embed patients' procedures in order to convert to numerical vectors.

Firstly, for each patient, we created a list containing the chronological procedures he or she did from his or her all hospital visits. The following table illustrates the procedure information in the dataset, the information in the *Short Title* column is the exact procedure done by patients and the corresponding codes are in the *ICD9_CODE* column.

Patient ID	ICD9_CODE	Short Title
109	5523	Closed renal biopsy
113	3612	Aortocor bypas-2 cor art

Generally, both patients' procedures and embedding sentences can be seen as sequences. In addition, each procedure and each word in embedding sentences has a specific meaning along with temporal relationships. So, we used *Word2Vec* as our embedding method to treat 'procedures' as 'words', 'patients' as 'sentences' capturing the temporal relationships and co-occurrence windows.

Since we have plenty of medical terms which are infrequent in the patient's procedure history, we chose to use Skip Gram for the embedding because Skip Gram is found to represent rare words well, compared to CBOW. We set the embedding dimension to be represented as 128 and have no minimum frequency.

Parameter Setting for Skip Gram Embedding

vector size	128
training algorithm	skip gram
min count	1
epoch	20
window	1

In general, each patient's number of procedures is different. However, since we would like to apply CNN and RNN for our predicting models, but both CNN and RNN require strict formats of inputs, so we created a random uniform padding array to pad for patients who have fewer procedures, in order to make sure that each patient(sentence) has a same number of procedures(words), which equals the number of procedures from the patient who has most procedures.

	largest number of procedure of the patient as sentence length					
sentence 1	128-number array	128-number array	128-number array	128-number array	padding array	padding array
sentence 2	128-number array	128-number array	128-number array	padding array	padding array	padding array
sentence 3	128-number array	padding array	padding array	padding array	padding array	padding array

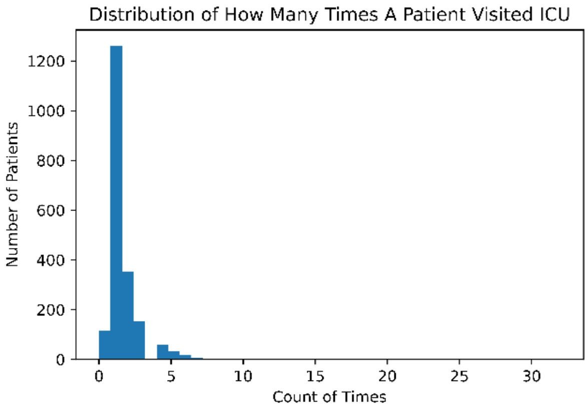
In addition, it is worth mentioning that we followed the work of Choi et al. (2016) to drop duplicate procedures in a patient's every visit for the embedding matrix, in order to achieve a good performing word embedding.

3.4 Exploratory Data Analysis

We have total of 2,020 patients in our dataset. The general features for these patients are *ICU times*, *length of stays*, *ICU types,* and *gender*, four histograms depict the distribution of these features for the 2,020 patients.

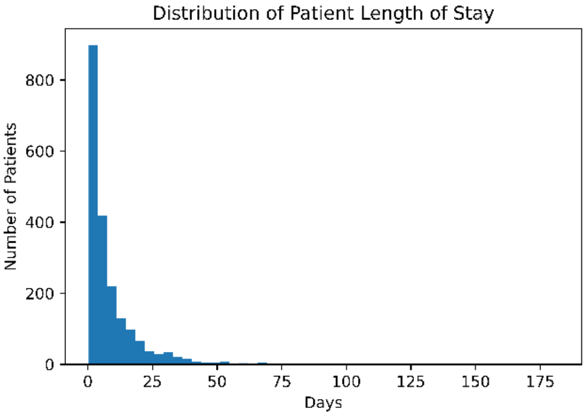
3.4.1 ICU Times

In our dataset, each patient visited ICU from 0 to 32 times, the figure below shows the distribution of how many times a patient visited ICU, most patients visited ICU 1 or 2 times.



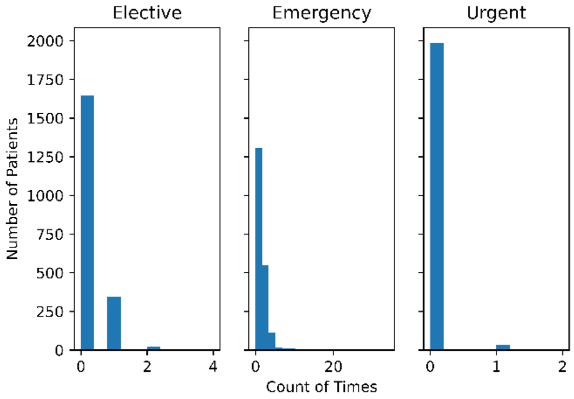
3.4.2 Length of Stays

Further, patients' lengths of stay are various depending on their own health conditions in ICU. In the length of stay histogram, we see that patients' cumulative length of stay in ICU, which is summation days for a patient's length of stay from his or her all ICU visits. The majority of patients stayed in ICU for less than 5 days, a small fraction of patients stayed in ICU for 5 to 20 days, only a few of patients stayed in ICU for 20 days.



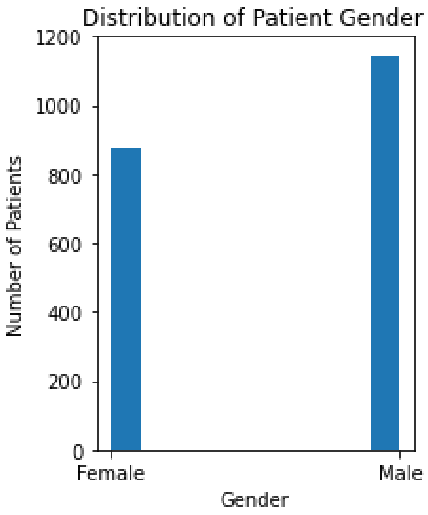
3.4.3 Admission Types

Admission type histogram shows how many times patients admitted by Elective, Emergency, and Urgent. Most patients admitted by Elective zero or once, nearly all patients were not admitted by Urgent, and the majority of patients admitted by Emergency from zero up to five times.



3.4.4 Gender

Gender histogram shows we have more male patients than female patients.



4. Deep Learning Models

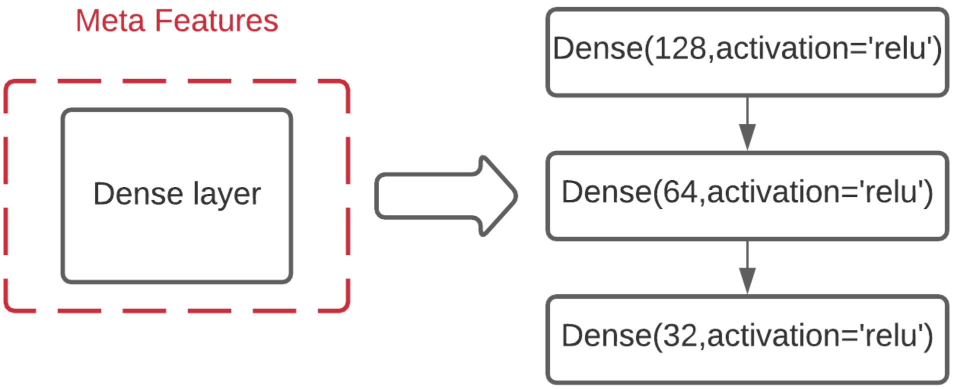
4.1 Overview of the Models and Metrics

The model architecture we chose to classify the mortality of patients is an infusion model. In the infusion models, we will have different “sub-architectures” to extract useful information in the features and concatenate the result to produce the prediction in patients’ mortality. After fine-tuning the model with proper hyperparameters, we will use the learning curve to diagnose the fitting process and use the classification recall, precision, f1-score, ROC curve, 10-fold cross-validation to compare the model performance and select the best model.

4.2 Architecture of Models

4.2.1 Dense Layers for Meta Feature

Our dense layer will have the following structure:

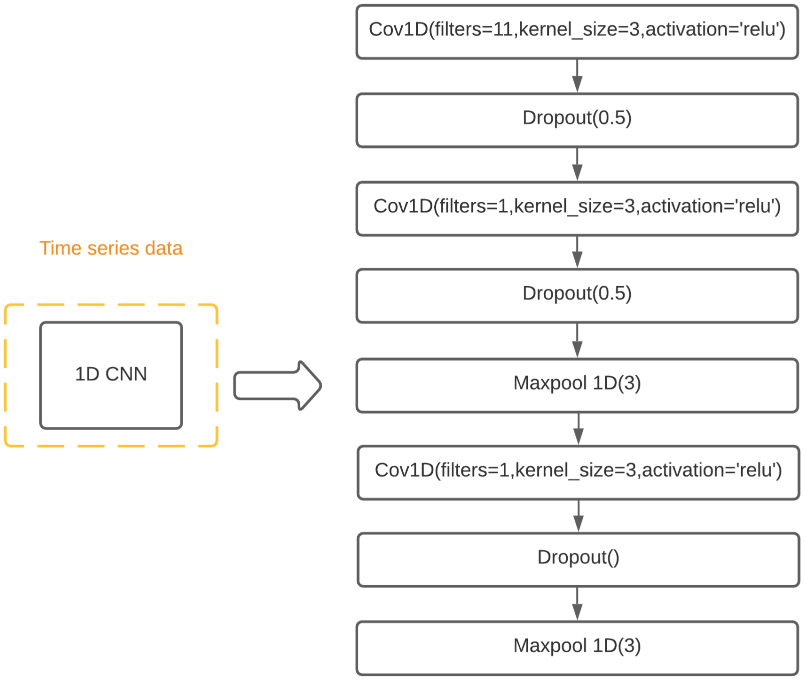

Meta Features
Dense layer
Dense(128,activation='relu')
Dense(64,activation='relu')
Dense(32,activation='relu')

The input features of the dense layers are single numerical observations with no special structure in the underlying data. To be more specific, the input features of the dense layer are:

- Admission type
- Gender of patient
- ICU times
- Length of stays
- Chart event series' mean and variance

4.2.2 Extracting the Features in *Labevents* Time Series: 1D CNN Layers

The 1D CNN layers will have the following structure:


Time series data
1D CNN
Cov1D(filters=11,kernel_size=3,activation='relu')
Dropout(0.5)
Cov1D(filters=1,kernel_size=3,activation='relu')
Dropout(0.5)
Maxpool 1D(3)
Cov1D(filters=1,kernel_size=3,activation='relu')
Dropout()
Maxpool 1D(3)

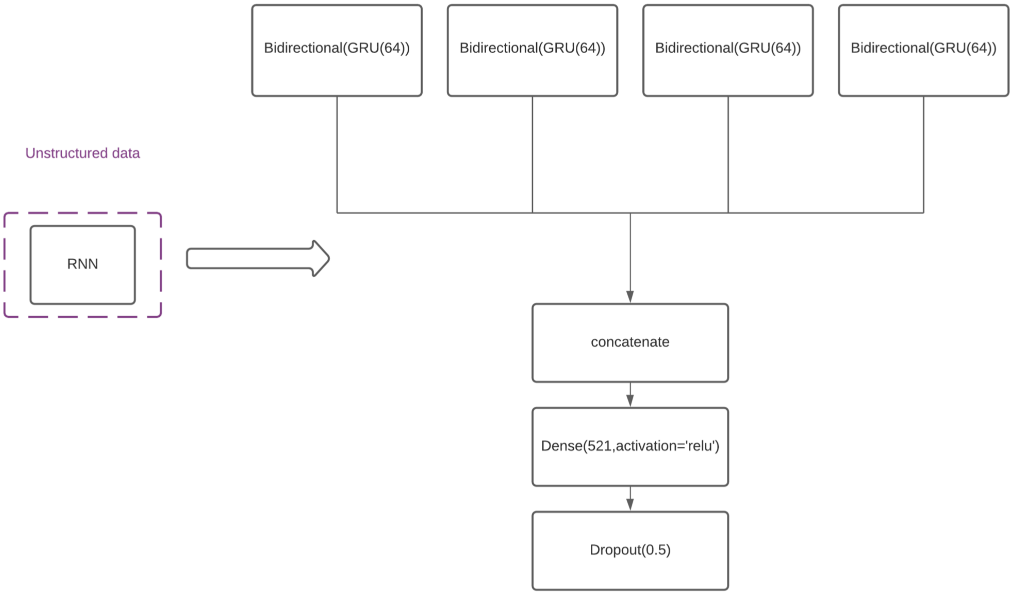
The kernel of the 1D CNN layers will be responsible for extracting the features in the *Labevents* time series. We believe the different lab items are not independent and should be considered simultaneously. One pattern in a lab item may have an influence on another lab item and the diagnosis of the patients is determined by the synergy of many lab results.

So, the 1D CNN layer would be a proper choice to process the lab event time series data. The input channels are the time series for different lab items and the kernel of the 1D CNN will convolve on all channels at one time. The kernel will move along in the time direction and will help us to extract the interactions between series.

4.2.3 Extracting Features in Unstructured Data: RNN and CNN Layers

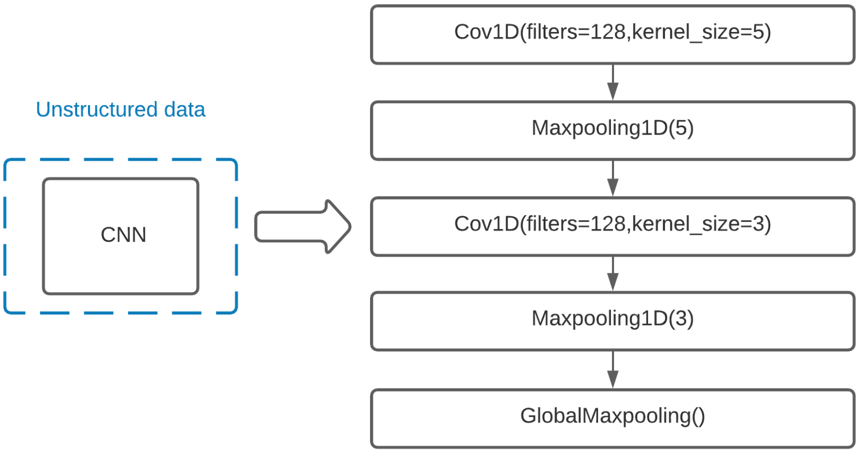
As mentioned before, we treated the medical procedure a patient took as a sequence and used the *word2vec* model to obtain the embedding vector of the patients' medical procedures. Also, we used the *doc2vec* model to obtain the numerical vectors to represent the patients' history of present illness, social history, brief hospital courses. To extract the pattern, we applied RNN layers and CNN layers respectively.

The RNN layers will have the following structure:



In the RNN layers, we treat the procedure embedding vector and the medical note document vector as a sequence. We use bidirectional GRU neurons to process the sequences. The memory cell in the GRU will help to consider the context of the input sequence (in the procedure embedding vector, the impact of previous medical treatment will be connected to the latter treatment. In the document vector, the context in the document will be correlated). With the bidirectional option, the GRU neurons are able to consider the further information that occurred later in the sequence.

The CNN layers will have the following structure:



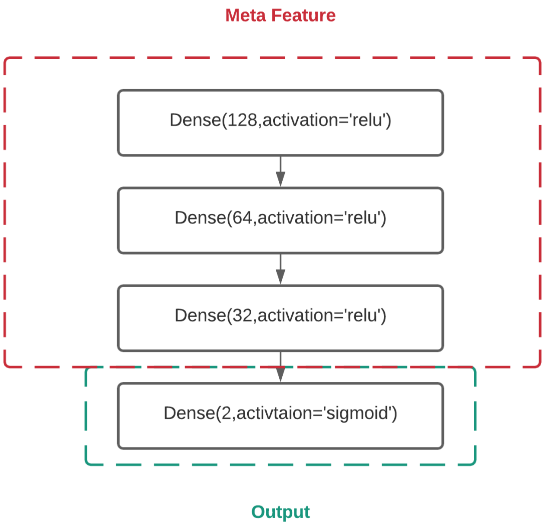
The kernel of the CNN layers will convolve between different procedure embedding vectors or medical note document vectors; the method is very similar to the Zhang, Y., & Wallace, B. (2015).

The intuition behind the method is that we believe the correlation between sequence data is very important. An effect of a procedure may be closely correlated to the previous one. The patient's medical note history may also have a correlation with the current medical note. By applying the CNN kernel in sequence, the important interactions of features will be extracted from embedding vectors or document vectors.

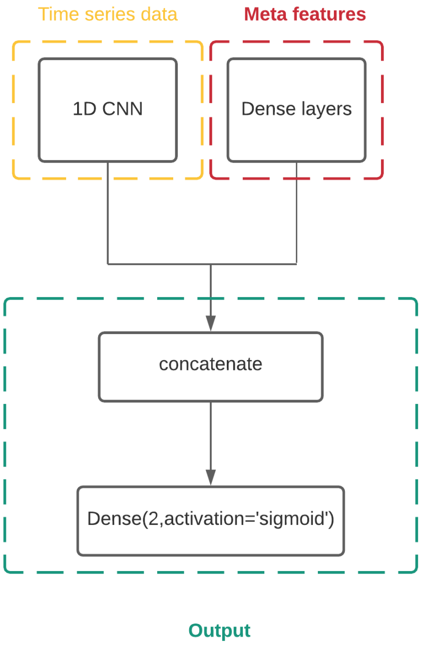
4.2.4 Infusion Model

Combining the different layers mentioned before, we will have four models:

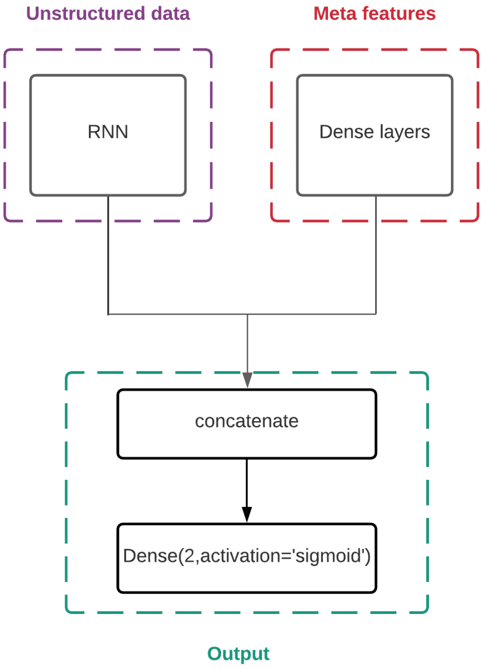
ANN model:


Meta Feature
Dense(128,activation='relu')
Dense(64,activation='relu')
Dense(32,activation='relu')
Dense(2,activtaion='sigmoid')
Output

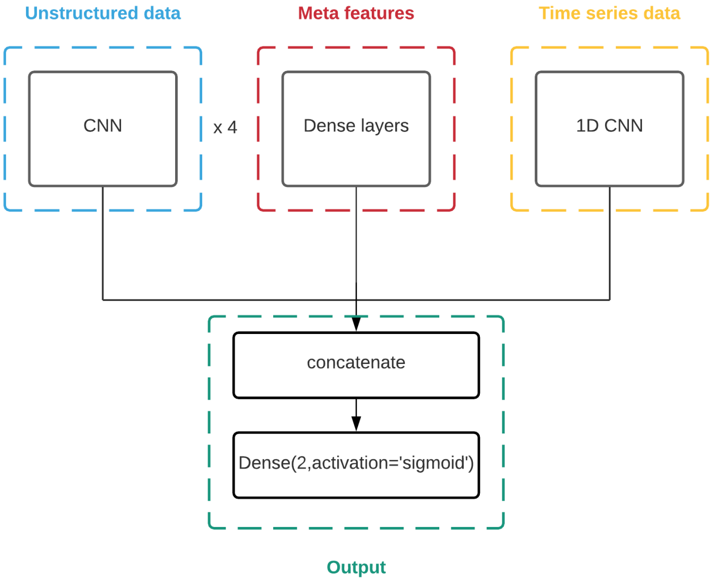
Fusion CNN + ANN model:



Fusion CNN + ANN + RNN model:



Fusion CNN + ANN + CNN model:

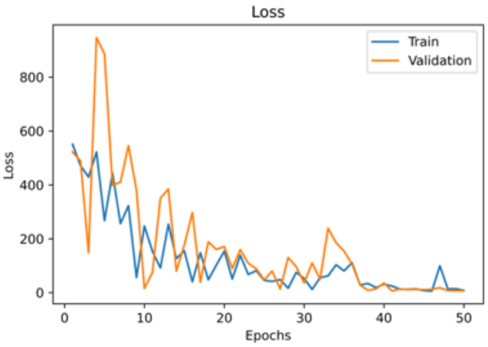


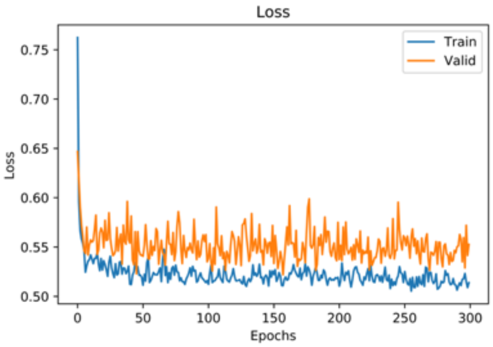
For different models, we will have:

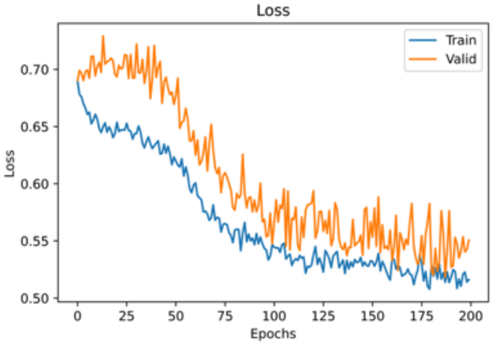
1. The ANN model will only consider the numerical data and it will not rely on the time-series data(lab events time series) and the unstructured data(medical procedure, medical note). It will be regarded as our baseline model.
2. The fusion CNN + ANN model will consider the numerical data and the time series data but will ignore the unstructured data. We use this model to determine whether the unstructured data will help to improve the model performance.
3. The fusion CNN + ANN + RNN and fusion CNN + ANN + RNN use different ways to process the unstructured data. With the model performance report of two models, we can compare which structure is more suitable to process the embedding vector and word(doc) vectors.

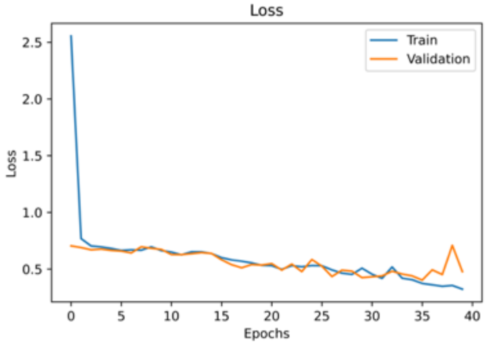
4.3 Model Performance

4.3.1 Learning Curve



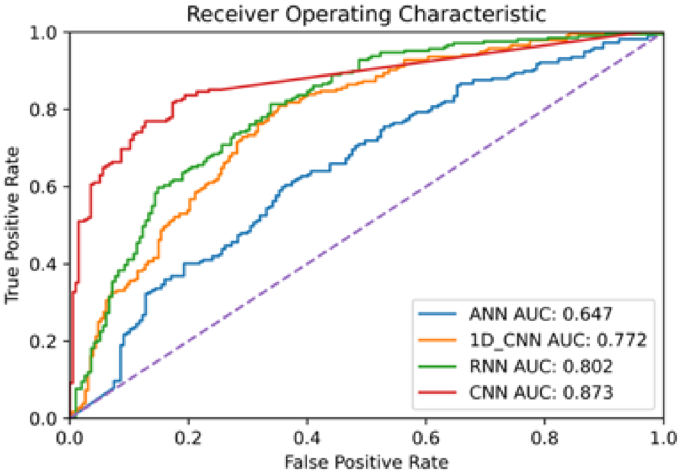






The four graphs show the learning curve for ANN,1D CNN, RNN and CNN. As we can see, all the models are not overfitted during the training process. As we increase the number of epochs, validation loss will gradually converge.

4.3.2 The ROC Curves



The graphs present the ROC curve of ANN, 1D CNN, RNN, and CNN. According to the graph, CNN has the highest AUC, which is 0.873. The RNN does second-highest, with a score of 0.802. When compared to 1D CNN, the ANN underperforms the 1D CNN but outperforms the random classifier.

4.3.3 The Classification Report

	Precision (Deceased)	Precision (Alive)	Precision (Average)	Recall	F1-Score
ANN	0.54	0.65	0.58	0.59	0.58
1D CNN	0.75	0.69	0.72	0.72	0.72
RNN	0.68	0.77	0.75	0.72	0.72
CNN	0.75	0.88	0.81	0.82	0.81

The table shows the performance of models in our project. CNN model has the greatest average precision, recall, and f1-score among other models, which is 0.81, 0.82, and 0.81, respectively. Compared with RNN, CNN is more robust than RNN, which means CNN can powerfully capture the pattern of unstructured data. Although 1D CNN is less accurate than RNN, 1D CNN shares the same recall and f1-score with RNN. Due to the lack of unstructured data and time-series data, ANN underperforms 1D CNN, RNN, and CNN.

4.3.4 The 10 Folds Cross-Validation Accuracy

	Mean acc	Var acc	Mean loss	Var loss
ANN	0.60	0.0014	15.80	369.664
1D CNN	0.75	0.0094	0.54	0.00056
RNN	0.73	0.0044	0.54	0.0043
CNN	0.83	0.0125	0.80	0.00041

The table demonstrates the 10-fold cross-validation results for ANN,1D CNN, RNN, and CNN. CNN gives us the best performance and its average accuracy can reach 0.83, however it achieves the highest variance during the cross-validation process, which is

0.0125. Compared with RNN, CNN can provide more accurate prediction results than RNN, which means RNN is not as effective as CNN to extract useful information from unstructured data to explain the patient's mortality. Compared with CNN, ANN is not as efficient as CNN to capture useful information to interpret patients' mortality.

5. Conclusion and Future Steps

From the previous analysis, we can see the CNN is the model with the best performance and the fusion models are effective ways in the mortality prediction task. The 1D CNN are able to extract the patterns from time-series data and the CNN module is an excellent structure in processing embedding vectors and document vectors.

If time permits, the project could be further extended in the following direction:

1. Early-stage prediction: Our current model needs to consider the patient's whole history to generate a mortality prediction. Even though the model accuracy is relatively high, it does not have much practical meaning. At the very late stage of the patient's hospital stay, it may not be hard to predict patient mortality because the critical condition is reelected on the patient obviously, and an experienced and trained doctor can easily predict the mortality of the patient. So, it is worth considering only to use the early-stage data to make the prediction, and the result is more beneficial to the healthcare companies.
2. A better way to extract the information in the medical note: Our current method ignored the entity's connection within the text, which may be a vital point to get the accuracy prediction. For example, a certain disease may be closely related to another disease and affect the posterior distribution of the patient getting this disease. Simply converting the medical history to document vectors will cause

information loss. To preserve as much information as possible when converting the text to numerical, an entity recognition in different diseases may be a helpful way.

Reference

JaWanna Henry et al. Adoption of Electronic Health Record Systems among U.S. Non-Federal Acute Care Hospitals: 2008-2015". en. In: (), p. 11.

Choi, Youngduck, Chill Yi-I Chiu, and David Sontag. 2016. "Learning Low-Dimensional Representations of Medical Concepts." *AMIA Summits on Translational Science Proceedings* 2016 (July): 41–50.

Youngduck Choi et al. Learning Low-Dimensional Representations of Medical Concepts". In: AMIA Summits on Translational Science Proceedings 2016 (July 2016), pp. 41{50. issn: 2153-4063. url: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5001761/ (visited on 03/05/2021).

Lance De Vine et al. Medical Semantic Similarity with a Neural Language Model". en. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. Shanghai China: ACM, Nov. 2014, pp. 1819{1822. isbn: 978-1-4503-2598-1. doi:10.1145/2661829.2661974. url: https://dl.acm.org/doi/10.1145/2661829.2661974 (visited on 03/05/2021).

Ning Ding et al. An Articial Neural Networks Model for Early Predicting In-Hospital Mortality in Acute Pancreatitis in MIMIC-III". en. In: BioMed Research International 2021 (Jan. 2021). Ed. by Dominique Monlezun, pp. 1{8. issn: 2314-6141, 2314-6133. doi: 10.1155/2021/6638919. url: https://www.hindawi.com/journals/bmri/2021/6638919/ (visited on 03/07/2021).

Shu Yuan Hu et al. An Explainable CNN Approach for Medical Codes Prediction from Clinical Text". en. In: arXiv:2101.11430 [cs] (Jan. 2021). arXiv: 2101.11430. url: http://arxiv.org/abs/2101.11430 (visited on 03/07/2021).

Dongdong Zhang et al. Combining structured and unstructured data for predictive models: a deep learning approach". en. In: BMC Medical Informatics and Decision Making 20.1 (Dec. 2020), p. 280.issn: 1472-6947. doi: 10.1186/s12911- 020-01297- 6. url: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-01297-6 (visited on 03/07/2021).

Zhang, Ye, and Byron Wallace. n.d. "A Sensitivity Analysis of (and Practitioners Guide to) Convolutional Neural Networks for Sentence Classification," 11.

Appendix A

SQL code to extract data:

```
# Lab Summary: 1
select ITEMID, LABEL, FLUID, CATEGORY, LOINC_CODE
   from `canvas-hook-308423.MIMIC.D_LABITEMS`
where LABEL in
('Anion gap', 'Albumin', 'Bands', 'Bicarbonate', 'Bilirubin', 'Creatinine', 'Chloride', 'Glucose', 'Hematocrit',
        'Hemoglobin', 'Lactate', 'Platelet', 'Potassium', 'PTT', 'INR', 'PT', 'Sodium', 'BUN', 'WBC')
order by LABEL;
```

```
# Lab Events: use paper measurements: 2
select A.Subject_id, L.ITEMID as ItemID, VALUE as Value, CHARTTIME as Time
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.LABEVENTS` L
where A.SUBJECT_ID =  L.SUBJECT_ID and L.ITEMID in (select ITEMID from
        `canvas-hook-308423.MIMIC_processed.D_LABITEMS`) order by A.SUBJECT_ID;
```

```
# vital sign: 3
select ITEMID, LABEL, DBSOURCE, LINKSTO,
   from `canvas-hook-308423.MIMIC.D_ITEMS`
where LABEL in
('Heart Rate', 'Non Invasive Blood Pressure systolic', 'Non Invasive Blood Pressure diastolic', 'Non Invasive
        Blood Pressure mean', 'Respiratory Rate', 'SpO2')
UNION ALL
select ITEMID, LABEL, DBSOURCE, LINKSTO,
   from `canvas-hook-308423.MIMIC.D_ITEMS`
where LABEL like '%Temperature%'
order by LABEL;
```

```
# vital sign: 4
select A.Subject_id, L.ITEMID as ItemID, VALUE as Value, CHARTTIME as Time
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.CHARTEVENTS` L
where A.SUBJECT_ID =  L.SUBJECT_ID and L.ITEMID in (select ITEMID from
        `canvas-hook-308423.MIMIC_processed.D_ITEMS`) order by A.SUBJECT_ID;
# Demographic: gender: 5
select A.Subject_id, P.GENDER as Gender, P.DOB as DOB
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.PATIENTS` P
where A.SUBJECT_ID =  P.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# META:
# length of stay: 6
select A.Subject_id, I.FIRST_CAREUNIT as First_unit, I.LAST_CAREUNIT as Last_unit, I.INTIME as
        Intime, I.OUTTIME as Outtime, i.LOS as LOS
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.ICUSTAYS` I
where A.SUBJECT_ID =  I.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# ICU times: 7
select A.Subject_id, C.CALLOUT_STATUS as Callout_status, C.CALLOUT_OUTCOME as
        Callout_outcome, C.OUTCOMETIME as Outcome_time
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.CALLOUT` C
where A.SUBJECT_ID =  C.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# Admission type: 8
select A.Subject_id, D.ADMISSION_TYPE as Admission_type
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.ADMISSION` D
where A.SUBJECT_ID =  D.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# Embedding
# Precsriptions: 9
select A.Subject_id, P.STARTDATE as Start_date, P.ENDDATE as End_date, P.DRUG_TYPE as
        Drug_type, P.DRUG_NAME_POE as Drug_name_poe, P.DRUG_NAME_GENERIC as
        Drug_name_generic, P.DOSE_VAL_RX as Dose_val, P.DOSE_UNIT_RX as Dose_unit
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.PRESCRIPTIONS` P
where A.SUBJECT_ID =  P.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# Procedures: 10
select A.Subject_id, P.STARTTIME as Start_time, P.ENDTIME as End_time, P.ITEMID as Item_id,
        P.VALUE as Value, P.VALUEUOM as Value_unit, P.STATUSDESCRIPTION as Status_des
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.PROCEDUREEVENTS_MV` P
where A.SUBJECT_ID =  P.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# service
select A.Subject_id, S.TRANSFERTIME as Transfer_time, S.CURR_SERVICE as Curr_service
   from `canvas-hook-308423.MIMIC_processed.ADMISSION_freq` A,
        `canvas-hook-308423.MIMIC.SERVICES` S
where A.SUBJECT_ID =  S.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# admission type
select A.Subject_id, D.ADMISSION_TYPE as Admission_type
    from `canvas-hook-308423.MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.ADMISSION_TYPE` D
where A.SUBJECT_ID =  D.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# chartevent
select A.Subject_id, D.ItemID, D.Value, D.Time
    from `canvas-hook-308423..MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.CHARTEVENTS` D
where A.Subject_id = D.Subject_id order by A.Subject_id;
```

```
# gender
select A.Subject_id, D.gender, D.DOB
    from `canvas-hook-308423..MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.GENDER` D
where A.Subject_id = D.Subject_id order by A.Subject_id;
```

```
# ICU times
select A.Subject_id, D.Callout_status, D.Callout_outcome, D.Outcome_time
    from `canvas-hook-308423..MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.ICU_TIMES` D
where A.Subject_id = D.Subject_id order by A.Subject_id;
# labevents
select A.Subject_id, L.ITEMID as ItemID, VALUE as Value, Time
    from `canvas-hook-308423.MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.LABEVENTS` L
where A.SUBJECT_ID =  L.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# los
select A.Subject_id, L.First_unit, L.Last_unit, L.Intime, L.Outtime, L.LOS
    from `canvas-hook-308423.MIMIC_final.INDEX` A, `canvas-hook-308423.MIMIC_processed.LOS` L
where A.SUBJECT_ID =  L.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# prescriptions
select A.Subject_id, L.Start_date, L.End_date, L.Drug_type, L.Drug_name_poe, L.Drug_name_generic,
           L.Dose_val, L.Dose_unit
    from `canvas-hook-308423.MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.PRESCRIPTIONS` L
where A.SUBJECT_ID =  L.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# procedure
select A.Subject_id, L.Start_time, L.End_time, L.Item_id, L.Value, L.Value_unit, L.Status_des
    from `canvas-hook-308423.MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.PROCEDURES` L
where A.SUBJECT_ID =  L.SUBJECT_ID order by A.SUBJECT_ID;
```

```
# service
select A.Subject_id, L.Transfer_time, L.Curr_service
    from `canvas-hook-308423.MIMIC_final.INDEX` A,
           `canvas-hook-308423.MIMIC_processed.SERVICE` L
where A.SUBJECT_ID =  L.SUBJECT_ID order by A.SUBJECT_ID;
```