Machine Learning Nanodegree Capstone Project: Mortality Predictor using MIMIC-III Database

Maxim V. Ivanov

I. BACKGROUND

Over the last decade, there has been a rising interest in adopting large medical databases to store patient records, data from clinical trials as well as medical research data. The ongoing big-data revolution in health care is largely driven by the benefits it brings to patients, physicians and stakeholders. In particular, a traditional approach where physicians used their judgement to make treatment decisions is being replaced by a data-oriented algorithmic approaches. In order for these algorithms to work most effectively, aggregation of individual medical data sets into big-data platforms is of outmost importance. A representative example of a large medical database is the Medical Information Mart for Intensive Care (MIMIC-III) database.[1] The MIMIC-III database is freely available and has been widely used in building predictive models as well as in epidemiological and educational studies.[2,3] Among various models that can be built using the information available in MIMIC-III is the mortality predictor of the patients admitted to the intensive care units (ICUs). Availability of accurate models trained on health care data, including mortality prediction, is critically important as it provide health providers with useful review of patient health upon their admission to the hospital, at the bedside and at the discharge. In this work, we construct a neural network model that predicts the mortality outcome of the patients admitted to the ICU based on various vital and lab measurements taken within first 24 hours upon admission. We discuss in detail how the training data has been obtained and preprocess along with the details on how the model was chosen and parameterized. We demonstrate that the constructed model performs significantly better than the empirical score-based models such as Acute Physiological Score (APS).[4] We also compare performance the constructed model with those discussed in the literature.

II. PROBLEM STATEMENT

One of the central issues in the data-oriented medical research is reproducibility of medical studies. Therefore, the goal of this project is twofold: on one hand, we demonstrate how neural networks can be used to build accurate models using health care data on the example of mortality predictor and, on the other hand, we focus on the reproducibility of our work. To this end, we employ the MIMIC Code Repository—a centralized location of concepts related to the critical care research. This will ensure that the training set contains features that represent a domain knowledge and enable reproducibility of the work.

III. DATASET

Here we use MIMIC-III v1.4 database, a freely-available dataset containing deidentified information on more than 60,000 stays in ICUs at the Beth Israel Deaconess Medical Center in Boston, Massachusetts between 2001 and 2012. The database includes information on patients' demographics, vital measurements, laboratory tests, medications, procedures, caregiver notes as well as survival data. In order to access the database, we completed a CITI "Data or Specimens Only Research" course that included Health Insurance Portability and Accountability Act (HIPAA) requirements and signed an agreement outlining appropriate data usage and security standards. MIMIC-III database contains a detailed demographics and medical data on each patient who stayed at the ICUs and, as such, requires domain knowledge in order to perform feature engineering that is relevant for mortality prediction. To this end, we utilize MIMIC Code Repository that provides detailed descriptions on how various relevant concepts are defined and extracted from the database.

IV. BENCHMARK MODEL AND METRICS

We start with the logistic regression classifier and move on to more complex models based on neural networks with several hidden layers. As the benchmark model we use the score-based APS model, i.e., a commonly used empirical model to estimate the severity of a patient's condition that is calculated using a set of vital and lab measurements.[4] To allow a fair comparison between our models and APS, the logistic regression and neural network models are built on a set of features that is similar to that used in the APS model. We then compare our results with

a recent study by Purushothama and co-workers, where mortality predictor was built using various deep learning models.[5]

To evaluate peformance of our models we rely on the Area Under the Receiver Operator Characteristic Curve (AUROC)—a common metric in the mortality prediction research. AUROC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

V. EXPLORATORY ANALYSIS

Before proceeding to the model development, we perform exploratory analysis in order to develop a better understanding of the data. We first examine demographics data that includes features like patients' age, weight, height, marital status, etc. For example, Figure 1 below shows the distribution of the patients' age. The distribution shows three distinct clusters: a sharp distribution at the age of 0, a binomial distribution for patients in the range of 14 to 89 with the mean of 62.1 years and another sharp distribution with the maximum at 300 years. Further analysis shows that out of a total of 61,532 admitted patients, 8,109 are newborns as indicated by the sharp distribution at the age of 0, and 2,722 are patients above age of 89. In order to comply with HIPAA and obscure the age of old patients, the date of birth of patients who are older than 89 years was shifted to exactly 300 years before their first admission.

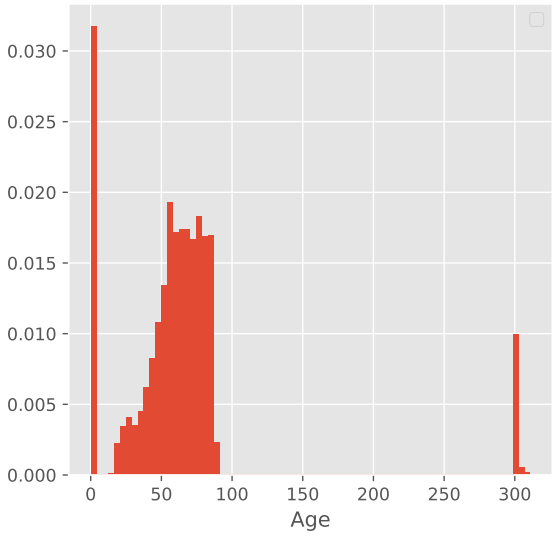

FIG. 1: Distribution of the patients' age.

We next examine the patients' medical data obtained at the bedside, including heart rate, blood pressure, respiration rate, body temperature, etc. For example, Figure 2, panel A shows the heart rate measurements during the entire period of state for a patient who survived (blue line) and compare with the measurements of a patient who did not survive at ICU (red line). Overall, these two measurements show drastically different patterns, suggesting that this feature is extremely important in identifying the mortatlity outcome. Indeed, a zoom-in into the first 24 hour period (Figure 2, panel B) shows that these two particular patients show a contrasting pattern in their heart rate measurements: blue line shows a much less varience than the red line. We therefore extract minimum, maximum and mean values of the heart rate and use them as features. Similarly, we analyze the signal obtained from the measurements of blood pressure, respiratory rate, body temperature, glucose level, etc. In majority of the cases we found that minimum, maximum and mean values of the measurements are expected to be very promising features that can help in distingushing the mortality outcome.

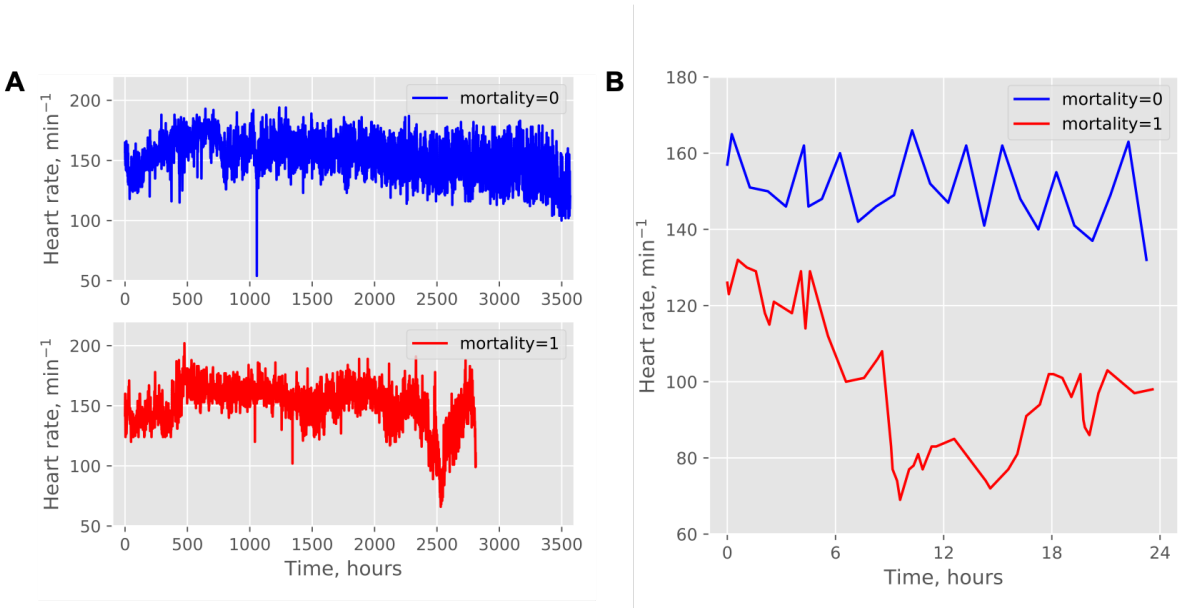

FIG. 2: Heart rate measurements during entire period of stay (A) and within first 24 hours (B) of representative patients who did (blue line) and did not (red line) survived at ICU.

Another interesting measurement that is somewhat distinct from the vital signals is the measurement of the urine output. The total urine output grows linearly with time as shown on the example of a representative patient in Figure 3. While using minimum, maximum and mean values of urine output as features is possible, are more reasonable feature would be a total volume of the urine output per day. The urine output of a healthy person is in the range

of 800-2,000 ml per day, therefore anything that falls outside of this range would signal of underlying health issues.

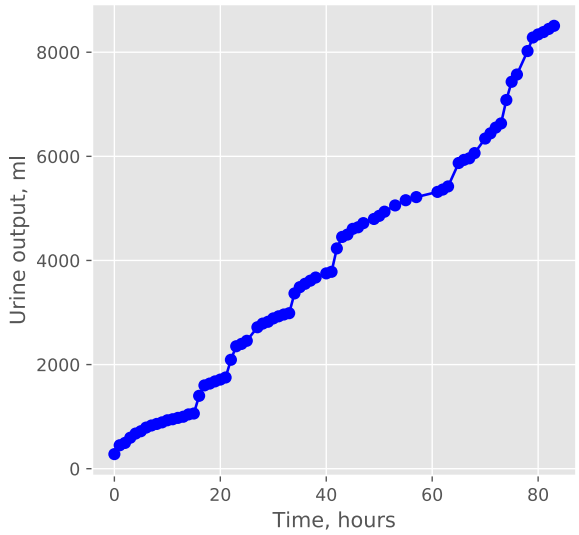

FIG. 3: Urine output measurements of a representative patient.

Next we examine the distribution of the per-day urine output across all patients and the histogram of the distribution is shown in Figure 4, panel A. While the majority of the instances lie within the range of 0-8,000 ml, there are multiple cases where the urine output takes extreme values up to 561,190 ml. While it seems like these extreme values are unphysically large, a domain expertise is required in order to draw a line between the values that are extreme but humanly possible and those that are humanly impossible and arise due to the error. Importantly, an extreme urine output value may indicate a severe health condition and may be important in identifying the mortality outcome. Therefore, we decided to shift all values that are beyond 10,000 ml to a constant value of 10,000 ml, which was determined empirically as a reasonable value. The distribution of the urine output after pre-processing is shown on the panel B in Figure 4. The distribution is positively skewed with the mean value of 1802 ml, which falls in the range of the healthy values. There is a number of patients that have a near-zero urine output, which, most likely, indicate a critical condition of the patient.

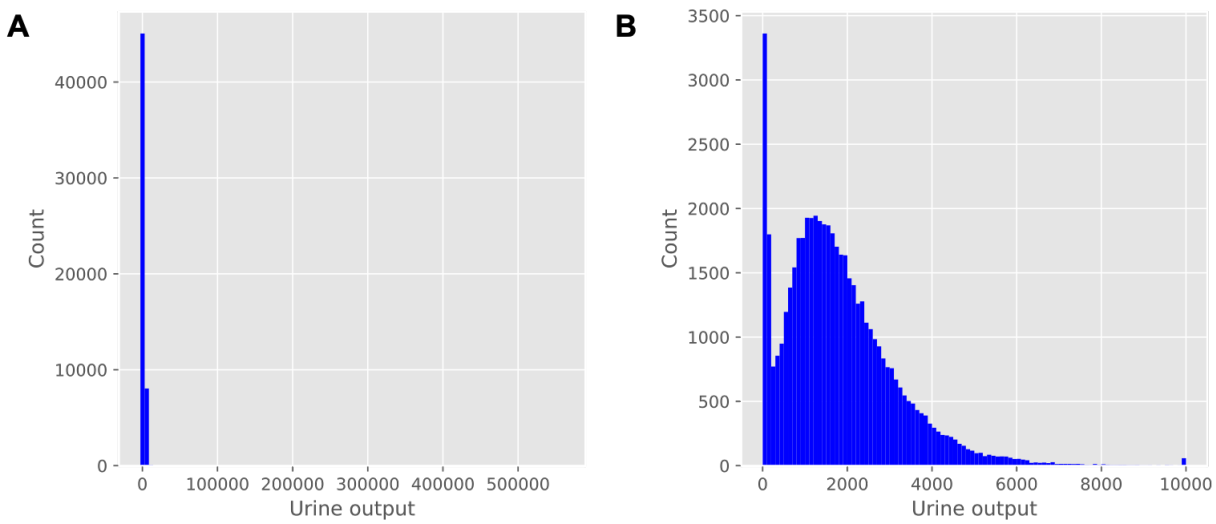

FIG. 4: Distribution of the urine output using unprocessed (A) and pre-processed (B) data.

VI. METHODOLOGY

A. Algorithms and benchmarks

From the exploratory analysis above it is clear that the mortality prediction is a complex problem and therefore an algorithm of our model should be chosen with care. In particular, mortality prediction is a non-linear problem and therefore we cannot expect that such models as logisic regression would be appropriate. Nevertheless, we can use logistic regression as the benchmark model to which we can compare more sophisticated models. In addition, we use the APS model as an additional benchmark because of its widespread usage in the field. As our model of choice we use feed-forward neural networks with several hidden layers because it allows identifying underlying relationships between a set of vital and lab measurements in a non-linear fashion. We will use the AUROC as the metric to compare performance of the models because it is a common metric in the clinical research.

B. Data preprocessing

Overall, our dataset contains 28 features and 61,532 unique admissions to ICU. Features were selected in a such way that they are comparable to those utilized in the APS model and include a set vital and lab measurements and their derivatives such as minimum, maximum and

mean values. Majority of the features originate from the medical device records and therefore missing values are inevitable. Figure 5 below shows the fraction of the missing values for each feature in the dataset. Clearly, the fraction of missing values is dependent on the origin of the data, yet majority of the features have at most 17% of missing data. Therefore, we are not dropping any features and impute all the missing values with the median value for each feature.

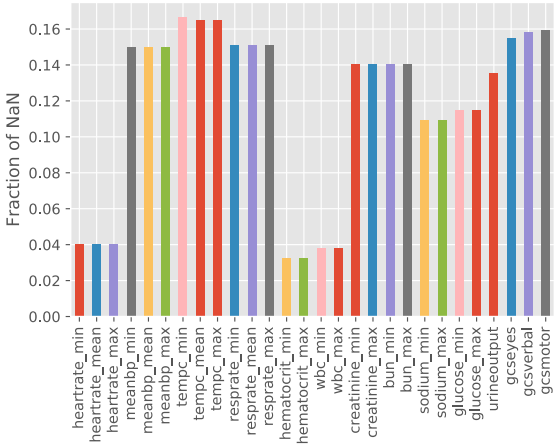

FIG. 5: Fraction of the missing data (NaN) in the dataset for each feature.

Similarly to the case of urine output discussed above, some of the features have extreme values that fall out of the range of normal values. However, given that we are not domain experts we cannot judge whether these outliers are humanly impossible and arise due to the error. Since some of the algorithms may be sensitive to the extreme values, we want to eliminate them and therefore constrain some of the features within an emprically adjusted range. For example, all glucose values were constrained to the maximum of 1,000 mg/dL as this values is a large enough to indicate that there are underlying issues, yet is not too far from the normal range of values. Once all features are adjusted to a reasonable range, we use a MinMaxScaler to bring all values within the range of 0 to 1.

C. Implementation

Majority of the admitted patients survive and the in-hospital mortality is relatively low, around 12%. This suggests that the dataset is imbalanced and, unless we do something about it, our model will tend to classify most cases as negative simply because it is a more frequently occuring class. At the same time, we prefer to minimize the number of false negatives (i.e., prediction that a patient survives, while in reality he/she does not) but may allow some false

positives (i.e., prediction that a patient does not survive, while in reality he/she does survive). Therefore, we want our model to have a high recall and we may allow a lower precision. Therefore, in addition to AUROC we will use precision, recall and their harmonic mean, i.e., F1 score, as additional metrics to judge the model performance.

In order to manage the class imbalance during training, we adjust weights of the classes inversely proportional to the frequencies of their occurance. Using this approach, the loss function is penalized more if the less frequent class is misclassified and therefore reducing the effect of class imbalance. Additionally, we use stratified sampling when splitting the dataset into the training (80%) and test (20%) sets in order to make sure that the class frequencies are same in training and test sets. To perform a hyper-parameter tuning we further split our training set into the sets used for training (80%) and validation (20%).

We begin with implementing logistic regression model as the simplest classifier using `LogisticRegression` class as implemented in `sklearn` with L-BFGS optimization algorithm and other parameters set to their default values. While other algorithms such as ensemble methods based on decision trees (e.g., random forest, adaptive or gradient boosting) are good candidates to be tried as well, here we are interested in the performance of neural networks.

To build neural networks we use the infrastructure available in `PyTorch` with all the training done locally on a MacBook Pro. We construct a feed-forward neural network with 28 units in the input layer, several hidden layers with varied number of units and the output layer with a single unit and sigmoid activation function that provides a probability of the mortality outcome. We employ ReLU activation function for the units in hidden layer and use dropout with hyper-parameter of 0.2 after each hidden layer to avoid overfitting. We use the Adam optimizer with learning rate of 0.001, batch size of 200, 200 epochs and binary cross entropy as the loss function.

D. Hyper-parameter tuning

Two major hyper-parameters in the neural networks are the depth (i.e., number of hidden layers) and width (i.e., number of units in each layer). Other hyper-parameters such as learning rate, batch size and number of epochs are also important, however here we focus on the depth and width of the neural network. First, we examine the optimal number of units in a neural network with a single hidden layer and compare with the performance of the logistic regression,

which can be viewed as neural network with a single unit.

As number of the units in a hidden layer increases, the neural network model is expected by able to better generalize until the addition of more units does not provide any improvement or until it deteriorates the peformance due to overfitting. Figure 6 shows a plot of the loss on the validation set against the epochs during the training of neural networks with varied number of units in the hidden layer. As number of units in the hidden layer increases, the model is capable of converging to the lower loss until the number of units reaches 124, which shows a minor improvement over the model with 96 units.

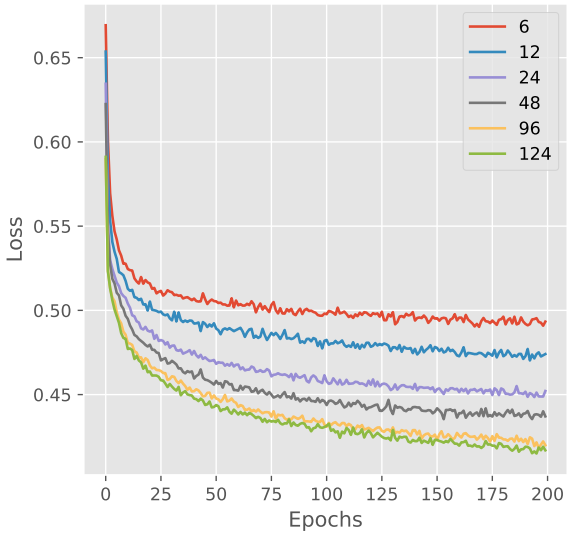

FIG. 6: A plot of the loss on the validation set against the epochs during the training of neural networks with varied number of units in the hidden layer.

Table I below shows performance measured using several metrics on a validation set for neural networks with varied number of units. Firstly, neural networks show a significant improvement over the simple logistic regression across all four metrics. Further increase of units in the hidden layer shows improvements in AUROC and F1, while precision and recall display a varience. The AUROC and F1 score converge at 96-124 units suggesting that 96 units is sufficient, which is consistent with earlier conclusion based on the convergence of the loss in Figure 6.

Next we examine how the depth of the neural network affects the model performance. Following models are constructed: one layer with 96 units (96), two layers with 48 units each (48,48), two layers with 64 and 32 units (64,32), two layers with 96 units each (96,96), three layers with 32 units each (32,32,32). Table II summarizes the performance of neural networks

with various architectures. Interestingly, a neural network with a single 96-unit hidden layer showed the best performance as measured by AUROC and F1 scores. All other architectures that included additional one or two hidden layers display a slightly deteriorated performance. This suggests that the non-linear transformations that occur in the single hidden layer are sufficient to identify underlying relationships between the vital measurements and predict the mortality outcome. This is in contrast to performance of neural networks in other domains such as image classification, where multiple hidden layers are required in order to identify higher-level patterns in the image.

TABLE I: Performance of neural network with a single hidden layer and various number of units. Model with a single unit corresponds to the logistic regression classifier.

# Units	AUROC	F1	Precision	Recall
1	0.840	0.400	0.280	0.740
6	0.861	0.438	0.306	0.766
12	0.868	0.429	0.292	0.809
24	0.875	0.452	0.318	0.783
48	0.877	0.458	0.323	0.788
96	0.882	0.471	0.338	0.777
124	0.882	0.445	0.305	0.818

TABLE II: Performance of neural networks with several hidden layers and various number of units.

Model	AUROC	F1	Precision	Recall
(96)	0.883	0.490	0.365	0.747
(48,48)	0.878	0.441	0.300	0.826
(64,32)	0.879	0.447	0.309	0.810
(96,96)	0.878	0.447	0.309	0.811
(32,32,32)	0.877	0.430	0.292	0.812

VII. RESULTS

Based on the discussion above our model of choice is a feed-forward neural network with a single hidden layer containing 96 units, which we compare below with the simple logistic regression and benchmark APS model. Figure 7 shows ROC curves for these three models and Table III summarizes their metrics. Across all three models, a neural network model shows the best performance as judged by all four metrics, i.e., AUROC, F1 score, precision and recall. It is interesting to note that APS, despite showing a relatively high AUROC of 0.799, has a

low recall and high precision. In contrast, both logistic regression and neural network models display much higher recall but lower precision. In many cases, it is hard to achieve both high recall and precision and therefore it is up to a developer to choose which of these two metrics is preferred. As discussed above here we aim to minimize the number of false negatives and therefore prefer a model that has a higher recall at the expense of lower precision.

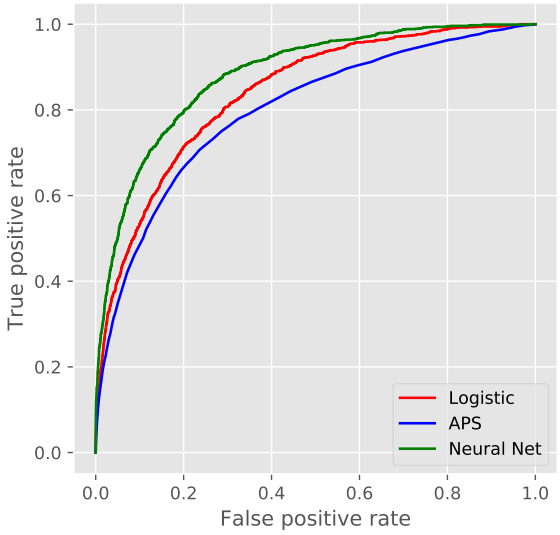

FIG. 7: ROC curves of the benchmark APS model (blue), logistic regression classifier (red) and neural network model (green).

TABLE III: Performance of the benchmark APS model, logistic regression classifier and neural network with one hidden layer.

Model	AUROC	F1	Precision	Recall
APS	0.799	0.229	0.651	0.139
Logistic regression	0.840	0.400	0.280	0.740
Neural network	0.883	0.490	0.365	0.747

Finally, we compare performance of our model with that of other deep learning models as reported by Purushotham and co-workers.[5] In their work, multiple machine learning models have been developed on feature sets of various sizes. The highest AUROC score of 0.941 was achieved using a deep learning model (hyper-parameters not specified) on a dataset with 135 features. A deep learning model trained on a set with 20 features that is comparable to that used in this work (28 features) achieved AUROC of 0.873, which is slightly less than the AUROC of 0.883 reported here.

In conclusion, in this work we constructed a neural network model that predicts the mortality outcome of a patient admitted to the ICU using the vital measurements data recorded within the first 24 hours of admission. We were particularly interested in how the neural network performs in comparison with more conventional approaches such APS, which follows an empirical rule to determine the likelihood of mortality outcome. Therefore, we selected a set of vital measurements that is comparable to that used in calculating APS and contained 28 features in total. We found that a neural network with single hidden layer and 96 units shows the best performance as has been judged using several different metrics. Our results show that neural network models are capable to achieve much higher predictive power as compared to standard empirical-based models.

---

[1] Johnson, Alistair EW; Pollard, Tom J; Shen, Lu; Li-wei, H Lehman; Feng, Mengling; Ghassemi, Mohammad; Moody, Benjamin; Szolovits, Peter; Celi, Leo Anthony; Mark, Roger G MIMIC-III, a freely accessible critical care database *Scientific data* **2016**, *3*, 160035.

[2] Hsu, Douglas J; Feng, Mengling; Kothari, Rishi; Zhou, Hufeng; Chen, Kenneth P; Celi, Leo A The association between indwelling arterial catheters and mortality in hemodynamically stable patients with respiratory failure: a propensity score analysis *Chest* **2015**, *148*, 1470–1476.

[3] Sun, JX; Reisner, AT; Saeed, M; Mark, RG Estimating cardiac output from arterial blood pressure-waveforms: a critical evaluation using the MIMIC II database *Computers in Cardiology* **2005**, page 295.

[4] Knaus, William A; Wagner, Douglas P; Draper, Elizabeth A; Zimmerman, Jack E; Bergner, Marilyn; Bastos, Paulo G; Sirio, Carl A; Murphy, Donald J; Lotring, Ted; Damiano, Anne et al. The apache iii prognostic system: risk prediction of hospital mortality for critically iii hospitalized adults *Chest* **1991**, *100*, 1619–1636.

[5] Purushotham, Sanjay; Meng, Chuizheng; Che, Zhengping; Liu, Yan Benchmark of deep learning models on large healthcare mimic datasets *arXiv preprint arXiv:1710.08531* **2017**.