The provided train set was split into train and validation sets (80/20). Once a model had been trained it was run against the validation set to assess its performance. This approach allowed submitting only reasonable models.

To train the model on the provided data different feature eng. approaches were tried. Firstly, patient data for each label was collapsed into one feature vector by computing the mean of available data points. If a data label had only NaN entries for all 12 hours the NaN was replaced by a zero. This led to results below 0.7 and was not useful (same results using median). Secondly, instead of zeroing the NaN values, these spots were filled with the mean values of available data of other patients. By using this approach the min. score of 0.71 was beaten with 0.723. Thirdly, missing values between data points were interpolated and missing values before the first available data entry were filled with this entry (e.g. time-step 2 has value 7.0, then time-steps 1 was filled with 7.0 of time-step 2, too). The same goes for the upper bound were values after the last data entry were filled with this last entry’s value. This avoided extrapolation. For columns with only NaN values we looked up reasonable values on the Internet and replaced the NaNs with them. Using this tuned data set, the score on the validation set was 0.755. For the test set a score of 0.739 was achieved. Another approach was to design one giant feature vector that stretches out all the 12 h feature vectors into one instead of computing a mean. This strategy did not improve the score further.

Tested models with promising scores were RandomForestClassifier (RFC), GradientBoostingClassifier (GBC), and HistGradientBoostingClassifer (HGBC) (regressors for the regression task). All 3 achieved scores > 0.71 on the validation set. RFC performed worst with 0.72. RFC and GBC both achieved > 0.75. However, HGBC was finally chosen as it runs about 2x faster than GBC (also stated in the doc).