Multitask learning and benchmarking with clinical time series data

Hrayr Harutyunyan[1], **Hrant Khachatrian**[2,3], **David C. Kale**[1], **Greg Ver Steeg**[1], **and Aram Galstyan**[1]

[1]USC Information Sciences Institute, Marina del Rey, California 90292, United States of America
[2]YerevaNN, Yerevan, 0025, Armenia
[3]Yerevan State University, Yerevan, 0025, Armenia
*hrant@yerevann.com

ABSTRACT

Health care is one of the most exciting frontiers in data mining and machine learning. Successful adoption of electronic health records (EHRs) created an explosion in digital clinical data available for analysis, but progress in machine learning for healthcare research has been difficult to measure because of the absence of publicly available benchmark data sets. To address this problem, we propose four clinical prediction benchmarks using data derived from the publicly available Medical Information Mart for Intensive Care (MIMIC-III) database. These tasks cover a range of clinical problems including modeling risk of mortality, forecasting length of stay, detecting physiologic decline, and phenotype classification. We propose strong linear and neural baselines for all four tasks and evaluate the effect of deep supervision, multitask training and data-specific architectural modifications on the performance of neural models.

Introduction

In the United States alone, each year over 30 million patients visit hospitals[1], 83% of which use an electronic health record (EHR) system[2]. This trove of digital clinical data presents a significant opportunity for data mining and machine learning researchers to solve pressing health care problems, such as early triage and risk assessment, prediction of physiologic decompensation, identification of high cost patients, and characterization of complex, multi-system diseases[3–7]. These problems are not new (the word *triage*, dates back to at least World War I and possibly earlier[8], while the Apgar risk score was first published in 1952[9]), but the success of machine learning[10,11] and growing availability of clinical data have sparked widespread interest.

While there has been a steady growth in machine learning research for health care, several obstacles have slowed progress in harnessing digital health data. The main challenge is the absence of widely accepted benchmarks to evaluate competing models. Such benchmarks accelerate progress in machine learning by focusing the community and facilitating reproducibility and competition. For example, the winning error rate in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) plummeted an order of magnitude from 2010 (0.2819) to 2016 (0.02991). In contrast, practical progress in clinical machine learning has been difficult to measure due to variability in data sets and task definitions[12–16]. Public benchmarks also lower the barrier to entry by enabling new researchers to start without having to negotiate data access or recruit expert collaborators.

Additionally, most of the researchers develop new methods for one clinical prediction task at a time (e.g., mortality prediction[12] or condition monitoring[17]). This approach is detached from the realities of clinical decision making, in which all the above tasks are often performed simultaneously by clinical staff[18]. Perhaps more importantly, there is accumulating evidence that those prediction tasks are interrelated. For instance, the highest risk and highest cost patients are often those with complex comorbidities[19] while decompensating patients have a higher risk for poor outcomes[5].

In this paper, we take a comprehensive approach to addressing the above challenges. We propose a public benchmark suite that includes four different clinical prediction tasks inspired by the opportunities for "big clinical data" discussed in Bates et al.[3]: in-hospital mortality, physiologic decompensation, length of stay (LOS), and phenotype classification. Derived from the publicly available Medical Information Mart for Intensive Care (MIMIC-III) database[20,21], our benchmark contains rich multivariate time series from over 40,000 intensive care unit (ICU) stays as well as labels for four tasks spanning a range of classic machine learning problems from multilabel time series classification to regression with skewed responses. These data are suitable for research on topics as diverse as non-random missing data and time series analysis.

This setup of benchmarks allows to formulate a *heterogeneous multitask learning* problem that involves jointly learning all four prediction tasks simultaneously. These tasks vary in not only output type but also temporal structure: LOS involves a regression at each time step, while in-hospital mortality risk is predicted once early in admission. Their heterogeneous nature

requires a modeling solution that can not only handle sequence data but also model correlations between tasks distributed in time. We demonstrate that carefully designed recurrent neural networks are able to exploit these correlations to improve the performance for several tasks.

Our code is already available online[78], so that anyone with access to MIMIC-III can build our benchmarks and reproduce our experiments sidestepping difficulties of preprocessing of clinical data.

Related Work

There is an extensive body of research on clinical predictions using deep learning, and we will attempt to highlight only the most representative or relevant work since a full treatment is not possible.

Feedforward neural networks nearly always outperform logistic regression and severity of illness scores in modeling mortality risk among hospitalized patients[22–24]. Recently, it was shown that novel neural architectures (including ones based on LSTM) perform well for predicting inpatient mortality, 30-day unplanned readmission, long length-of-stay (binary classification) and diagnoses on general EHR data (not limited to ICU)[25]. The experiments were done on several private datasets.

There is a great deal of early research that uses neural networks to predict LOS in hospitalized patients[26,27]. However, rather than regression, much of this work formulates the task as binary classification aimed at identifying patients at risk for long stays[28]. Recently, novel deep learning architectures have been proposed for survival analysis[29,30], a similar time-to-event regression task with right censoring.

Phenotyping has been a popular application for deep learning researchers in recent years, though model architecture and problem definition vary widely. Feedforward networks[31,32], LSTM networks[33] and temporal convolutional networks[34] have been used to predict diagnostic codes from clinical time series. In 2016, it was first shown that recurrent neural networks could classify dozens of acute care diagnoses in variable length clinical time series[35].

Multitask learning has its roots in clinical prediction[22]. Several authors formulated phenotyping as multi-label classification, using neural networks to implicitly capture comorbidities in hidden layers[34,35]. Others attempted to jointly solve multiple related clinical tasks, including predicting mortality and length of stay[36]. However, none of this work addressed problem settings where sequential or temporal structure varies across tasks. The closest work in spirit to ours is a paper by Collobert and Weston[37] where a single convolutional network is used to perform a variety of natural language tasks (part-of-speech tagging, named entity recognition, and language modeling) with diverse sequential structure.

Earlier version of this work has been available online for two years (arXiv:1703.07771v1). The current version adds more detailed description of the dataset generation process, improves the neural baselines and adds more discussion on the results. Since the release of the preliminary version of the benchmark codebase, several teams used our dataset generation pipeline (fully or partially). In particular, the pipeline was used for in-hospital mortality prediction[38–43], decompensation prediction[44], length-of-stay prediction[42,44,45], phenotyping[38,39,46] and readmission prediction[47]. Additionally, attention-based RNNs were applied for all our benchmark tasks[48].

In a parallel work another set of benchmark tasks based on MIMIC-III was introduced that includes multiple versions of in-hospital mortality predictions, length-of-stay and ICD-9 code group predictions, but does not include decompensation prediction[49]. The most critical difference is that in all their prediction tasks the input is either the data of the first 24 or 48 hours, while we do length of stay and decompensation prediction at each hour of the stay, and do phenotyping based on the data of the entire stay. We frame the length-of-stay prediction as a classification problem and use the Cohen's kappa score as its metric, while they frame it as a regression problem and use the mean squared error as its metric. The metric they use is less indicative of performance given that the distribution of length of stay has a heavy tail. In ICD-9 code group prediction, we have 25 code groups as opposed to their 20 groups. There are many differences in the data processing and feature selection as well. For example, we exclude all ICU stays where the patient is younger than 18, while they exclude patients younger than 15. Moreover, they consider only the first admission of a patient, while we consider all admissions. They have benchmarks for three different features sets: A, B, and C, while we have only one set of features, which roughly corresponds to their feature set A. The set of baselines is also different. While our work has more LSTM-based baselines, the parallel work has more baselines with traditional machine learning techniques.

Results

We compile a subset of the MIMIC-III database containing more than 31 million clinical events that correspond to 17 clinical variables listed in the first column of Table 3. These events cover 42276 ICU stays of 33798 unique patients. We define four benchmark tasks on this subset.

1. In-hospital mortality prediction – predicting in-hospital mortality based on the first 48 hours of an ICU stay. This is a binary classification task with area under the receiver operating characteristic (AUC-ROC) being the main metric.

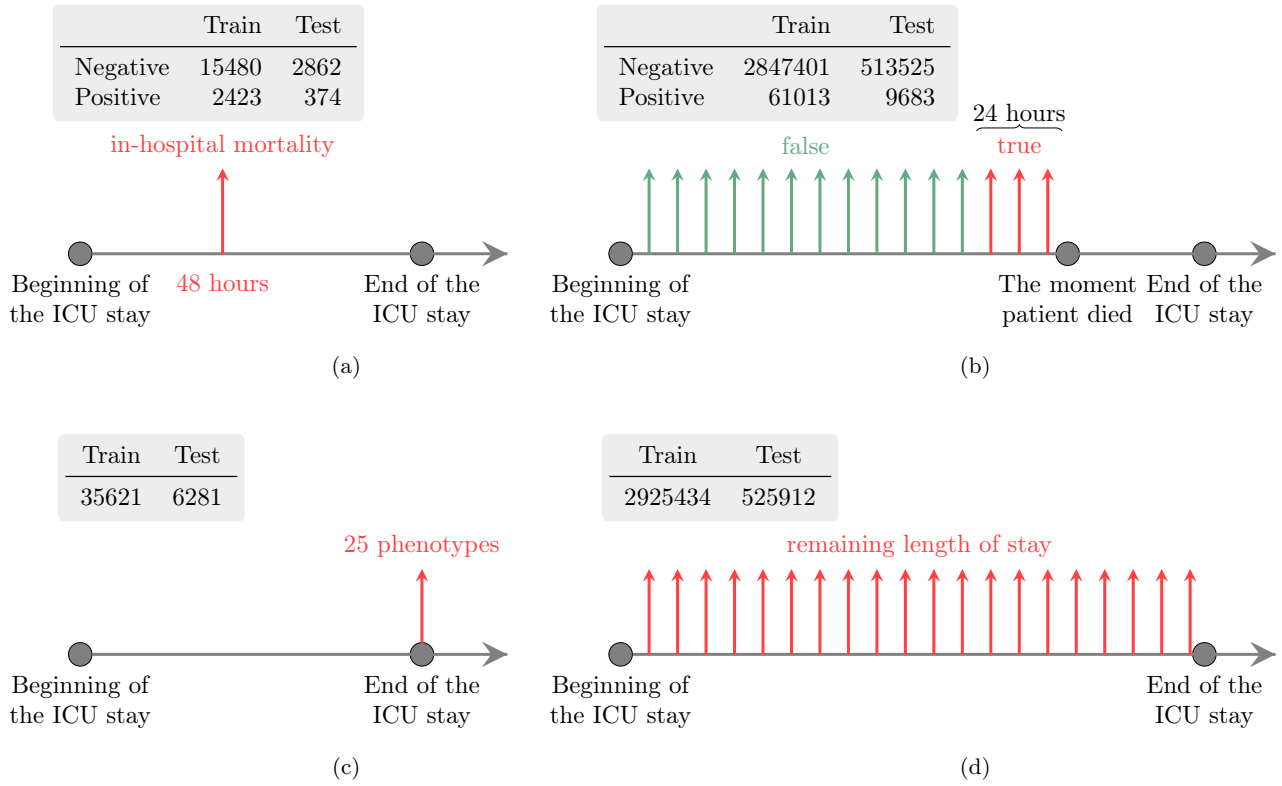


Figure 1. Summaries of the four benchmark tasks. Each subfigure consists of two parts. The table lists number of prediction instances for the corresponding task. The timeline shows when the predictions are done. Note that in the decompensation and length-of-stay prediction tasks predictions are done hourly, and each vertical arrow corresponds to one prediction instance. (a) In-hospital mortality. (b) Decompensation. (c) Phenotyping. (d) Length of stay.

2. Decompensation prediction – predicting whether the patient's health will rapidly deteriorate in the next 24 hours. The goal of this task is to replace early warning scores currently used in the hospitals. Due to the lack of gold standard for evaluating the early warning scores, we follow earlier work[5] and define our task as mortality prediction in the next 24 hours at each hour of an ICU stay. It is important to note that this definition deviates from the core meaning of decompensation, and the task becomes similar to the first one. On the other hand, we believe this is the closest proxy task for decompensation prediction for which one can obtain precise labels from MIMIC-III database. Each instance of this task is a binary classification instance. Likewise in-hospital mortality prediction, the main metric is AUC-ROC.

3. Length-of-stay prediction – predicting remaining time spent in ICU at each hour of stay. Accurate prediction of the remaining length-of-stay is important for scheduling and hospital resource management. We frame this as a classification problem with 10 classes/buckets (one for ICU stays shorter than a day, seven day-long buckets for each day of the first week, one for stays of over one week but less than two, and one for stays of over two weeks). The main metric for this task is Cohen's linear weighted kappa score.

4. Phenotype classification – classifying which of 25 acute care conditions (described in Table 2) are present in a given patient ICU stay record. This problem is a multilabel classification problem with macro-averaged AUC-ROC being the main metric.

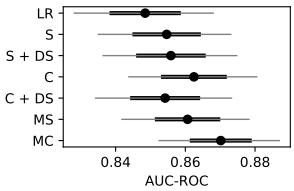
Additionally, a multitask version of the four tasks is defined. The tasks are summarized in the Figure 1.

We develop linear regression models and multiple neural architectures for the benchmark tasks. We perform experiments with a basic LSTM-based neural network (standard LSTM) and introduce a modification of it (channel-wise LSTM). Additionally, we test both types of LSTMs with deep supervision and multitask training. We perform a hyperparameter search to select the best performing models and evaluate them on the test sets of the corresponding tasks. By doing bootstrapping on the test set we also report 95% confidence intervals for the models and test the statistical significance of the differences between models. The results for each of the mortality, decompensation, LOS, and phenotyping tasks are reported in Figure 2.

LR – logistic regression
S – standard LSTM
C – channel-wise LSTM
DS – deep supervision
MS – multitask standard LSTM
MC – multitask channel-wise LSTM

In-hospital Mortality

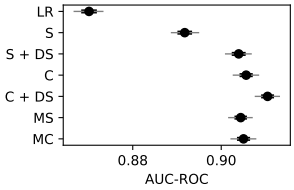
Model	AUC-ROC	AUC-PR
SAPS	0.720 (0.720, 0.720)	0.301 (0.301, 0.302)
APS-III	0.750 (0.750, 0.750)	0.357 (0.356, 0.357)
OASIS	0.760 (0.760, 0.761)	0.311 (0.311, 0.312)
SAPS-II	0.777 (0.776, 0.777)	0.376 (0.376, 0.377)
LR	0.848 (0.828, 0.868)	0.474 (0.419, 0.529)
S	0.855 (0.835, 0.873)	0.485 (0.431, 0.537)
S + DS	0.856 (0.836, 0.875)	0.493 (0.438, 0.549)
C	0.862 (0.844, 0.881)	0.515 (0.464, 0.568)
C + DS	0.854 (0.834, 0.873)	0.502 (0.447, 0.554)
MS	0.861 (0.842, 0.878)	0.493 (0.439, 0.548)
MC	0.870 (0.852, 0.887)	0.533 (0.480, 0.584)



	LR	S	S + DS	C	C + DS	MS	MC
LR	-	15.8	12.9	1.9	18.1	4.1	0.0
S	84.2	-	38.5	3.4	53.8	13.7	0.2
S + DS	87.1	61.5	-	10.7	63.4	19.2	0.6
C	98.1	96.6	89.3	-	95.5	62.3	4.7
C + DS	81.9	46.2	36.6	4.5	-	12.9	0.1
MS	95.9	86.3	80.8	37.7	87.1	-	0.9
MC	100.0	99.8	99.4	95.2	99.9	99.1	-

Decompensation

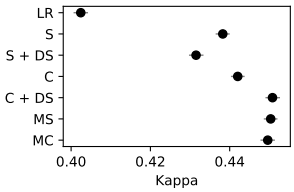
Model	AUC-ROC	AUC-PR
LR	0.870 (0.867, 0.873)	0.214 (0.205, 0.223)
S	0.892 (0.889, 0.895)	0.324 (0.314, 0.333)
S + DS	0.904 (0.901, 0.907)	0.325 (0.314, 0.335)
C	0.906 (0.903, 0.909)	0.333 (0.323, 0.344)
C + DS	0.911 (0.908, 0.913)	0.344 (0.334, 0.354)
MS	0.904 (0.902, 0.907)	0.321 (0.312, 0.331)
MC	0.905 (0.902, 0.908)	0.317 (0.307, 0.328)



	LR	S	S + DS	C	C + DS	MS	MC
LR	-	0.0	0.0	0.0	0.0	0.0	0.0
S	100.0	-	0.0	0.0	0.0	0.0	0.0
S + DS	100.0	100.0	-	6.2	0.0	35.1	13.2
C	100.0	100.0	93.8	-	0.0	91.4	73.1
C + DS	100.0	100.0	100.0	100.0	-	100.0	100.0
MS	100.0	100.0	64.9	8.6	0.0	-	12.5
MC	100.0	100.0	86.8	26.9	0.0	87.5	-

Length of stay

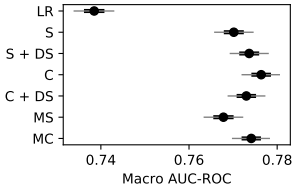
Model	Kappa	MAD
LR	0.402 (0.401, 0.404)	162.3 (161.8, 162.8)
S	0.438 (0.436, 0.440)	123.1 (122.6, 123.5)
S + DS	0.431 (0.430, 0.433)	110.9 (110.5, 111.4)
C	0.442 (0.440, 0.444)	136.6 (136.1, 137.1)
C + DS	0.451 (0.449, 0.453)	143.1 (142.6, 143.6)
MS	0.450 (0.449, 0.452)	112.0 (111.5, 112.5)
MC	0.450 (0.448, 0.451)	122.8 (122.3, 123.3)



	LR	S	S + DS	C	C + DS	MS	MC
LR	-	0.0	0.0	0.0	0.0	0.0	0.0
S	100.0	-	100.0	0.0	0.0	0.0	0.0
S + DS	100.0	0.0	-	0.0	0.0	0.0	0.0
C	100.0	100.0	100.0	-	0.0	0.0	0.0
C + DS	100.0	100.0	100.0	100.0	-	79.7	99.5
MS	100.0	100.0	100.0	100.0	20.3	-	91.0
MC	100.0	100.0	100.0	100.0	0.5	9.0	-

Phenotyping

Model	Macro AUC-ROC	Micro AUC-ROC
LR	0.739 (0.734, 0.743)	0.799 (0.796, 0.803)
S	0.770 (0.766, 0.775)	0.821 (0.818, 0.825)
S + DS	0.774 (0.769, 0.778)	0.823 (0.820, 0.827)
C	0.776 (0.772, 0.781)	0.825 (0.822, 0.828)
C + DS	0.773 (0.769, 0.777)	0.822 (0.819, 0.826)
MS	0.768 (0.763, 0.772)	0.818 (0.815, 0.822)
MC	0.774 (0.770, 0.778)	0.823 (0.819, 0.826)



	LR	S	S + DS	C	C + DS	MS	MC
LR	-	0.0	0.0	0.0	0.0	0.0	0.0
S	100.0	-	0.1	0.0	1.7	97.6	0.2
S + DS	100.0	99.9	-	1.3	68.7	100.0	36.5
C	100.0	100.0	98.7	-	99.6	100.0	96.9
C + DS	100.0	98.3	31.3	0.4	-	100.0	20.1
MS	100.0	2.4	0.0	0.0	0.0	-	0.0
MC	100.0	99.8	63.5	3.1	79.9	100.0	-

Figure 2. Results for in-hospital mortality, decompensation, length-of-stay, and phenotype prediction tasks. A subfigure corresponding to a benchmark task has three parts. The first part is a table that lists the values of the metrics for all models along with 95% confidence intervals obtained by resampling the test set K times with replacement ($K = 10000$ for in-hospital mortality and phenotype prediction task, while for decompensation and length-of-stay prediction tasks $K = 1000$). For all metrics except MAD, larger values are better. The second part visualizes the confidence intervals for the main metric of the corresponding task. The black circle corresponds to the mean value of K iterations. The thick black line shows standard deviation and narrow grey line shows 95% confidence interval. The third part shows the significance of the difference between the models. We count the number of resampled tests sets on which the i-th model performed better than the j-th model (denoted by $c_{i,j}$). The cell at the i-th row and the j-th column of the table shows the percentage of $c_{i,j}$ in K. We say that the i-th model is significantly better than the j-th model if $c_{i,j}/K > 0.95$ and highlight the corresponding cell of the table.

We first note that LSTM-based models outperformed linear models by substantial margins across all metrics on every task. The difference is significant in every case except three out of six LSTM models for in-hospital mortality. This is consistent with previous research comparing neural networks to linear models for mortality prediction[23], and phenotyping[35] but it is nonetheless noteworthy because questions still remain about the potential effectiveness of deep learning for health data, especially given the often modest size of the data relative to their complexity. Our results provide further evidence that complex architectures can be effectively trained on non-Internet scale health data and that while challenges like overfitting persist, they can be mitigated with careful regularization schemes, including dropout and multitask learning.

The experiments show that channel-wise LSTMs and multitask training act as regularizers for almost all tasks. Channel-wise LSTMs perform significantly better than standard LSTMs for all four tasks, while multitasking helps for all tasks except phenotyping (the difference is significant for decompensation and length-of-stay prediction tasks). We hypothesize that this is because phenotype classification is itself a multitask problem and already benefits from regularization by sharing LSTM layers across the 25 different phenotypes. The addition of further tasks with loss weighting may limit the multitask LSTM's ability to effectively learn to recognize individual phenotypes. Note that the hyperparameter search for multitask models did not include zero coefficients for any of the four tasks. That is why the best multitask models sometimes perform worse than single-task models.. The combination of the channel-wise layer and multitasking is also useful. Multitask versions of channel-wise LSTMs perform significantly better than the corresponding single-task versions for in-hospital mortality prediction and phenotyping tasks.

Deep supervision with replicated targets did not help for in-hospital mortality prediction. For phenotyping, it helped for the standard LSTM model (as discovered in an earlier work[35]), but did not help for channel-wise models. On the other hand, we see significant improvements from deep supervision for decompensation and length-of-stay prediction tasks (except for the Standard LSTM model for length-of-stay prediction). For both these tasks the winner models are channel-wise LSTMs with deep supervision. For decompensation, the winner is significantly better than all other models and for LOS the winner is significantly better than all others except the runner-up model, which is a multitask standard LSTM.

Discussion

In this paper we proposed four standardized benchmarks for machine learning researchers interested in clinical data problems, including in-hospital mortality, decompensation, length-of-stay, and phenotype prediction. Our benchmark data set is similar to other MIMIC-III patient cohorts described in machine learning publications but makes use of a larger number of patients and is immediately accessible to other researchers who wish to replicate our experiments or build upon our work. We also described several strong baselines for our benchmarks. We have shown that LSTM-based models significantly outperform linear models, although we expect to see better performing linear models by using more complex feature engineering. We have demonstrated the advantages of using channel-wise LSTMs and learning to predict multiple tasks using a single neural model. Our results indicate that the phenotyping and length-of-stay prediction tasks are more challenging and require larger model architectures than mortality and decompensation prediction tasks. Even small LSTM models easily overfit the latter two problems.

We note that since the data in MIMIC-III is generated within a single EHR system, it might contain systematic biases. It is an interesting future study to explore how models trained on these benchmarks generalize to other clinical datasets.

In-hospital mortality prediction

For risk-related tasks like mortality and decompensation, we are also interested how reliable the probabilities estimated by our predictive models are. This is known as *calibration* and is a common method for evaluating predictive models in the clinical research literature. In a well calibrated model, 10% all patients who receive a predicted 0.1 probability of decompensation do in fact decompensate. We included no formal measure of calibration in our benchmark evaluations, but we informally visualize calibration for mortality and decompensation predictions using reliability plots. These are scatter plots of predicted probability vs. actual probability. Better calibrated predictions will fall closer to the diagonal. Figure 3 (a) visualizes calibration of several in-hospital mortality prediction baselines. We see that the LSTM-based models look reasonably calibrated, while the logistic regression baseline consistently overestimates the actual probability of mortality. We would like to note that the logistic regression model can be successfully calibrated using Platt scaling or isotonic regression.

To provide more insights on the results of this task we present the receiver operating characteristic (ROC) curves corresponding to the best logistic, non-multitask and multitask LSTM-based in-hospital mortality prediction models in Figure 4 (a). Additionally, with Figure 5 (a) we demonstrate, perhaps unsurprisingly, that the performance of the best non-multitask baseline (channel-wise LSTM) degrades as length-of-stay increases. Finally, in the in-hospital mortality part of the Figure 2 we compare our baselines with the traditional scores like SAPS[50]. Note that these scores are computed on the first 24 hours of the stays.

Out of the four tasks, in-hospital mortality prediction task is the only one for which most of the differences between different baselines are not significant. The reason behind this is the relatively small size of the test set (about twice smaller than that of phenotype prediction task and much more smaller than those of length-of-stay and decompensation prediction tasks).

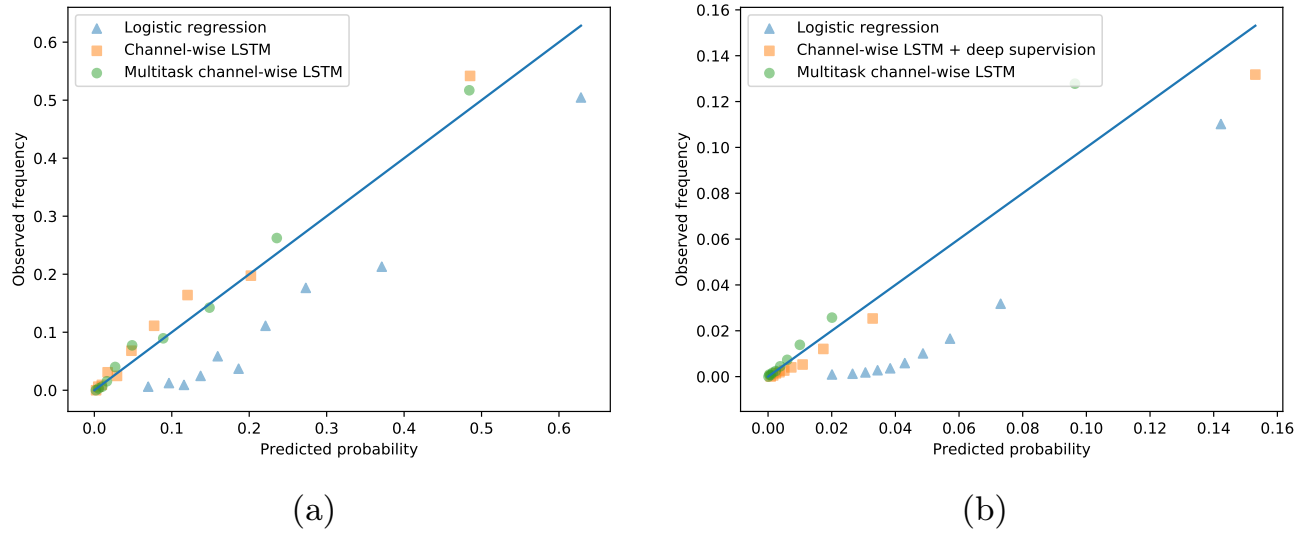


Figure 3. Calibration of in-hospital mortality and decompensation prediction by the best linear, non-multitask and multitask LSTM-based models. The plots show predicted probability computed by creating decile bins of predictions and then taking the mean value within each bin vs. actual probability (the rate of mortality within each bin). (a) In-hospital mortality. (b) Decompensation.

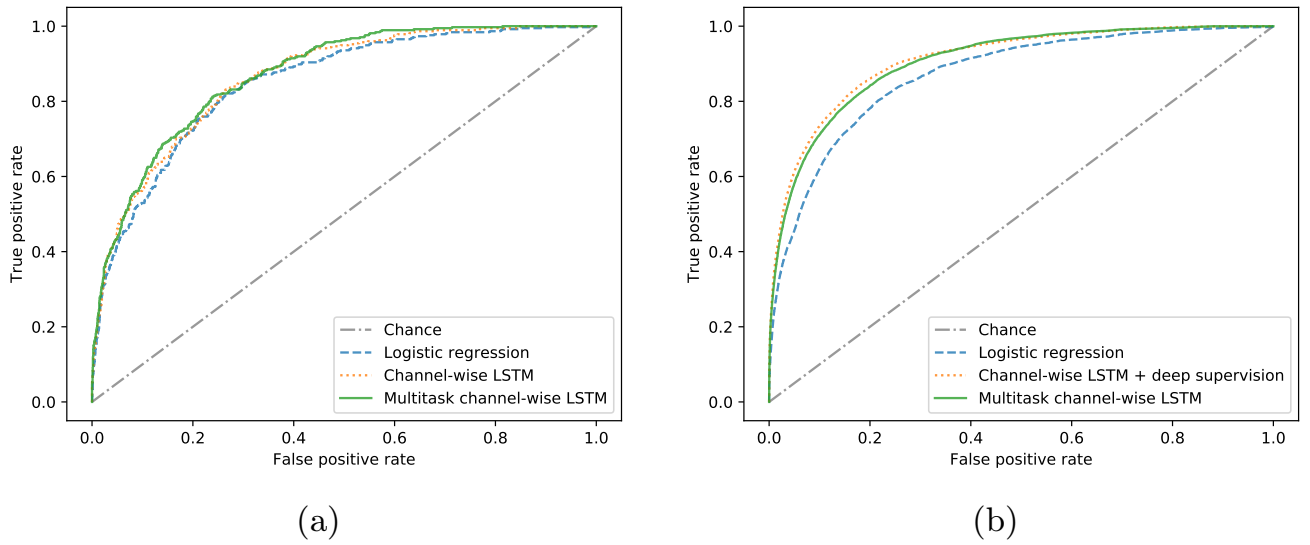
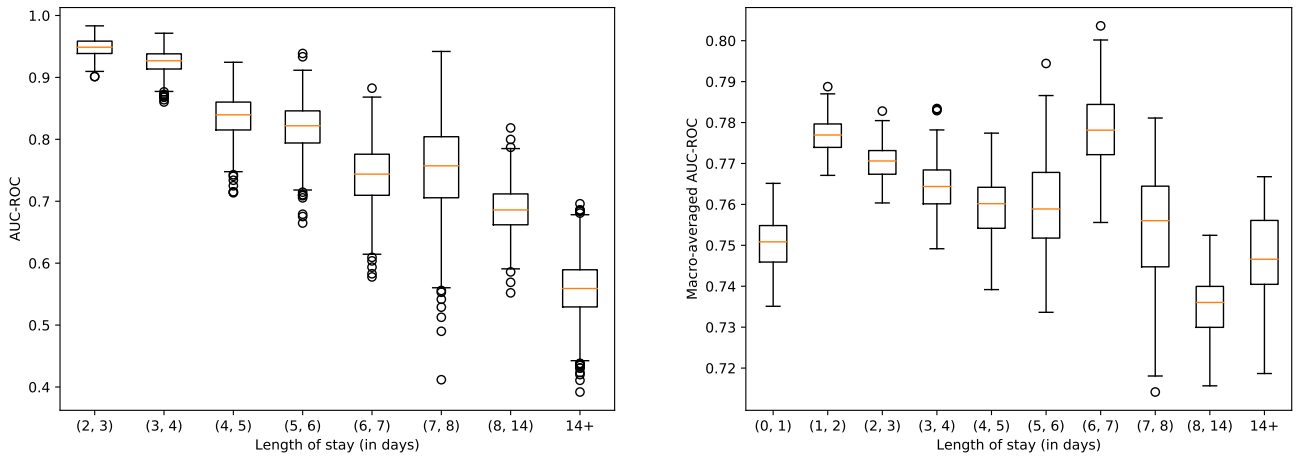


Figure 4. Receiver operating characteristic curves for the best linear, non-multitask and multitask LSTM-based models. (a) In-hospital mortality. (b) Decompensation

Decompensation prediction

Figure 3 (b) visualizes calibration of several decompensation prediction baselines. Likewise the case of in-hospital mortality prediction task, we see that the LSTM-based models are better calibrated than the logistic regression model. Again, the logistic regression model can be reasonably calibrated using Platt scaling or isotonic regression. Compared to in-hospital mortality baselines, we see that decompensation baselines are worse calibrated. This behavior is expected since the decompensation prediction has more severe class imbalance.

To provide further insights on the results we present the ROC curves corresponding to the best logistic, non-multitask and multitask LSTM-based decompensation prediction models in Figure 4 (b). Additionally, to understand better what the best decompensation prediction model (channel-wise LSTM with deep supervision) does, we visualize its predictions over the time in Figure 6. The left part of the figure shows randomly chosen 100 patients from the test set that died in ICU. The right part shows another 100 patients randomly chosen from the test set. Every row shows the predictions for the last 100 hours of a



(a) (b)

Figure 5. In-hospital mortality and phenotype prediction performance vs. length-of-stay. The plots show the performance of the best non-multitask prediction baselines on the test data of different length-of-stay buckets. The confidence intervals and standard deviations are estimated with bootstrapping on the data of each bucket. (a) In-hospital mortality. (b) Phenotype.

single ICU stay. Darker colors indicate higher predicted probability of death in the upcoming 24 hours. Red and blue colors indicate ground truth labels (blue is positive mortality). The right part of the figure shows that for living patients the model rarely produces false positives. The left part shows that in many cases the model predicts mortality days before the actual time of death. On the other hand, there are many cases when the mortality is predicted only in the last few hours, and in a few cases the model doesn't give high probability even at the last hour. This figure shows that even a model with 0.91 AUC-ROC can make trivial mistakes and there is a lot of room for improvement.

Length-of-stay prediction

For length-of-stay prediction task we also tried regression models that directly predict the number of days. These models consistently performed worse than classification models in terms of kappa score, but had better mean absolute difference (MAD), as shown in Table 1. Note that the regression models for LOS prediction were trained with mean squared error loss function. Therefore, the MAD scores they get are suboptimal and can be improved by directly training to minimize the MAD score. In general, our results for LOS forecasting are the worst among the four tasks. Our intuition is that this due in part to the intrinsic difficulty of the task, especially distinguishing between stays of, e.g., 3 and 4 days.

To investigate this intuition further, we considered a task formulation similar to the one described in Rajkomar et al.[25] where the goal was to predict whether a patient would have an extended LOS (longer than seven days) from only the first 24 hours of data. In order to evaluate our models in a similar manner, we summed the predicted probabilities from our multiclass LOS model for all buckets corresponding to seven days or longer LOS. For our best LOS model, this yielded an AUC-ROC of 0.84 for predicting extended LOS at 24 hours after admission. This is comparable to the results from Rajkomar et al. who reported AUC-ROCs of 0.86 and 0.85 on two larger private datasets using an ensemble of several neural architectures. This is especially noteworthy since our models were not trained to solve this particular problem and suggests that the extended LOS problem is more tractable than the regression or multiclass versions. Nonetheless, solving the more difficult fine-grained LOS problem remains an important goal for clinical machine learning researchers.

Phenotyping

Phenotyping is actually a combination of 25 separate binary classification tasks and the performance of the models vary across different tasks. Table 2 shows the per-phenotype ROC-AUC for the best phenotype baseline (channel-wise LSTM). We observe that AUC-ROC scores on the individual diseases vary widely from 0.6834 (essential hypertension) to 0.9089 (acute cerebrovascular disease). Unsurprisingly, chronic diseases are harder to predict than the acute ones (0.7475 vs 0.7964).

We did not detect any positive correlation between disease prevalence and ROC-AUC score. Moreover, the worst performance is observed for the most common phenotype (essential hypertension).

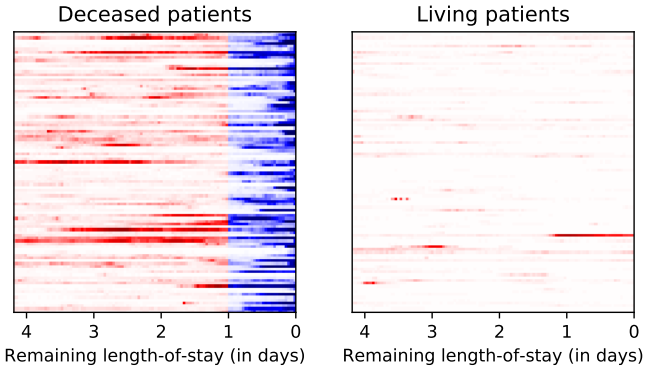


Figure 6. Prediction of channel-wise LSTM baseline with deep supervision for decompensation prediction over time. Each row shows the last 100 hours of a single ICU stay. Darker colors mean high probability predicted by the model. Red and blue colors indicate the ground-truth label is negative and positive, respectively. Ideally, the right image should be all white, and the left image should be all white except the right-most 24 hours, which should be all dark blue.

Multitask learning

We demonstrated that the proposed multitask learning architecture allows us to extract certain useful information from the input sequence that single-task models could not leverage, which explains the better performance of multitask LSTM in some settings. We did not, however, find any significant benefit in using multitask learning for the phenotyping task.

We are interested in further investigating the practical challenges of multitask training. In particular, for our four very different tasks, the model converges and then overfits at very different rates during training. This is often addressed through the use of heuristics, including a multitask variant of early stopping, in which we identify the best epoch for each task based on individual task validation loss. We proposed the use of per-task loss weighting, which reduced the problem but did not fully mitigate it. One promising direction is to dynamically adapt these coefficients during training, similar to the adaptation of learning rates in optimizers.

Methods

This section consists of three subsections. We describe the process of benchmark data and task generation along with evaluation metrics in the first subsection. The second subsection describes the linear and neural baseline models for the benchmark tasks. We describe the experimental setup and model selection in the third subsection.

Benchmark tasks

We first define some terminology: in MIMIC-III *patients* are often referred to as *subjects*. Each patient has one or more *hospital admissions*. Within one admission, a patient may have one or more *ICU stays*, which we also refer to as *episodes*. A clinical *event* is an individual measurement, observation, or treatment. In the context of our final task-specific data sets, we use the word *sample* to refer to an individual record processed by a machine learning model. As a rule, we have one sample for each prediction. For tasks like phenotyping, a sample consists of an entire ICU stay. For tasks requiring hourly predictions, e.g., LOS, a sample includes all events that occur before a specific time, and so a single ICU stay yields multiple samples.

Table 1. Results for length of stay prediction task (regression). Contrary to mean absolute deviation (MAD), larger kappa is better.

Model	Kappa	MAD
Linear regression	0.336 (0.335, 0.338)	116.4 (115.8, 117.0)
Standard LSTM	0.433 (0.432, 0.435)	94.7 (94.2, 95.1)
Standard LSTM + deep supervision	0.413 (0.411, 0.414)	94.5 (94.1, 95.0)
Channel-wise LSTM	0.424 (0.422, 0.425)	94.3 (93.9, 94.8)
Channel-wise LSTM + deep supervision	0.426 (0.424, 0.428)	94.0 (93.6, 94.4)

Table 2. ICU phenotypes used in the benchmark data set along with their prevalence and the per-phenotype classification performance of the best LSTM network

Phenotype	Type	Prevalence		AUC-ROC
		Train	Test	
Acute and unspecified renal failure	acute	0.214	0.212	0.806
Acute cerebrovascular disease	acute	0.075	0.066	0.909
Acute myocardial infarction	acute	0.103	0.108	0.776
Cardiac dysrhythmias	mixed	0.321	0.323	0.687
Chronic kidney disease	chronic	0.134	0.132	0.771
Chronic obstructive pulmonary disease	chronic	0.131	0.126	0.695
Complications of surgical/medical care	acute	0.207	0.213	0.724
Conduction disorders	mixed	0.072	0.071	0.737
Congestive heart failure; nonhypertensive	mixed	0.268	0.268	0.763
Coronary atherosclerosis and related	chronic	0.322	0.331	0.797
Diabetes mellitus with complications	mixed	0.095	0.094	0.872
Diabetes mellitus without complication	chronic	0.193	0.192	0.797
Disorders of lipid metabolism	chronic	0.291	0.289	0.728
Essential hypertension	chronic	0.419	0.423	0.683
Fluid and electrolyte disorders	acute	0.269	0.265	0.739
Gastrointestinal hemorrhage	acute	0.072	0.079	0.751
Hypertension with complications	chronic	0.133	0.130	0.750
Other liver diseases	mixed	0.089	0.089	0.778
Other lower respiratory disease	acute	0.051	0.057	0.694
Other upper respiratory disease	acute	0.040	0.043	0.785
Pleurisy; pneumothorax; pulmonary collapse	acute	0.087	0.091	0.709
Pneumonia	acute	0.139	0.135	0.809
Respiratory failure; insufficiency; arrest	acute	0.181	0.177	0.907
Septicemia (except in labor)	acute	0.143	0.139	0.854
Shock	acute	0.078	0.082	0.892
All acute diseases (macro-averaged)				0.796
All mixed (macro-averaged)				0.768
All chronic diseases (macro-averaged)				0.746
All diseases (macro-averaged)				0.776

Our benchmark preparation workflow, illustrated in Figure 7, begins with the full MIMIC-III critical care database, which includes over 60,000 ICU stays across 40,000 critical care patients. In the first step (`extract_subjects.py`), we extract relevant data from the raw MIMIC-III tables and organize them by patient. We also apply exclusion criteria to admissions and ICU stays. First, we exclude any hospital admission with multiple ICU stays or transfers between different ICU units or wards. This reduces the ambiguity of outcomes associated with hospital admissions rather than ICU stays. Second, we exclude all ICU stays where the patient is younger than 18 due to the substantial differences between adult and pediatric physiology. The resulting *root* cohort has 33,798 unique patients with a total of 42,276 ICU stays and over 250 million clinical events.

In the next two steps, we process the clinical events. The first step, `validate_events.py`, filters out 45 million events that cannot be reliably matched to an ICU stay in our cohort. First, it removes all events for which admission ID (HADM_ID) is not present. Then it excludes events that have admission ID that is not present in the `stays.csv` database which connects ICU stay properties (e.g. length of stay and mortality) to admissions IDs. Next, it considers events with missing ICU stay IDs (ICUSTAY_ID). For all such events the ICUSTAY_ID is reliably recovered by looking at the HADM_ID. Finally, the script excludes all events for which the ICUSTAY_ID is not listed in the `stays.csv` database. More detailed description of `validate_events.py` script can be found in the code repository[78].

The second step, `extract_episodes_from_subjects.py`, compiles a time series of events for each episode, retaining only the variables from a predefined list and performing further cleaning. We use 17 physiologic variables representing a subset from the Physionet/CinC Challenge 2012[51]. The selected variables are listed in the first column of Table 3. The data for each of these variables is compiled from multiple MIMIC-III variables. For example, there are eight variables in `chartevents`

table that correspond to weight (denoted by item IDs 763, 224639, 226512, 3580, 3693, 3581, 226531 and 3582). These variables come in different units (kg, oz, lb). We convert all values to kilograms. We do similar preprocessing for all 17 variables. The mapping of MIMIC-III item IDs to our variables is available at `resources/itemid_to_variable_map.csv` file of our code repository[78], where the first column is the name of the variable in our benchmark. The script `preprocessing.py` contains functions for converting these values to a unified scale.

The resulting data has over 31 million events from 42,276 ICU stays.

Variable	MIMIC-III table	Impute value	Modeled as
Capillary refill rate	chartevents	0.0	categorical
Diastolic blood pressure	chartevents	59.0	continuous
Fraction inspired oxygen	chartevents	0.21	continuous
Glascow coma scale eye opening	chartevents	4 spontaneously	categorical
Glascow coma scale motor response	chartevents	6 obeys commands	categorical
Glascow coma scale total	chartevents	15	categorical
Glascow coma scale verbal response	chartevents	5 oriented	categorical
Glucose	chartevents, labevents	128.0	continuous
Heart Rate	chartevents	86	continuous
Height	chartevents	170.0	continuous
Mean blood pressure	chartevents	77.0	continuous
Oxygen saturation	chartevents, labevents	98.0	continuous
Respiratory rate	chartevents	19	continuous
Systolic blood pressure	chartevents	118.0	continuous
Temperature	chartevents	36.6	continuous
Weight	chartevents	81.0	continuous
pH	chartevents, labevents	7.4	continuous

Table 3. The 17 selected clinical variables. The second column shows the source table(s) of a variable from MIMIC-III database. The third column lists the "normal" values we used in our baselines during the imputation step, and the fourth column describes how our LSTM-based baselines treat the variables.

Finally (`split_train_and_test.py`), we fix a test set of 15% (5,070) of patients, including 6,328 ICU stays and 4.7 million events. We encourage researchers to follow best practices by interacting with the test data as infrequently as possible. Finally, we prepare the task-specific data sets.

Our benchmark prediction tasks include four in-hospital clinical prediction tasks: modeling risk of mortality shortly after admission[4], real-time prediction of physiologic decompensation[5], continuous forecasting of patient LOS[6], and phenotype classification[35]. Each of these tasks is of interest to clinicians and hospitals and is directly related to one or more opportunities for transforming health care using big data[3]. These clinical problems also encompass a range of machine learning tasks, including binary and multilabel classification, regression, and time series modeling, and so are of interest to data mining researchers.

In-hospital mortality

Our first benchmark task involves prediction of in-hospital mortality from observations recorded early in an ICU admission. Mortality is a primary outcome of interest in acute care: ICU mortality rates are the highest among hospital units (10% to 29% depending on age and illness), and early detection of at-risk patients is key to improving outcomes.

Interest in modeling risk of mortality in hospitalized patients dates back over half a century: the Apgar score[9] for assessing risk in newborns was first published in 1952, the widely used Simplified Acute Physiology Score (SAPS)[50] in 1984. Intended to be computed by hand, these scores are designed to require as few inputs as possible and focus on individual abnormal observations rather than trends. However, in the pursuit of increased accuracy, such scores have grown steadily more complex: the Acute Physiology and Chronic Health Evaluation (APACHE) IV score requires nearly twice as many clinical variables as APACHE II[4].

Recent research has used machine learning techniques like state space models and time series mining to integrate complex temporal patterns instead of individual measurements[14,52]. Others leverage information from clinical notes, extracted using topic models[12,13]. These approaches outperform traditional baselines but have not been compared on standardized benchmarks.

Risk of mortality is most often formulated as binary classification using observations recorded from a limited window of time following admission. The target label indicates whether the patient died before hospital discharge. Typical models include

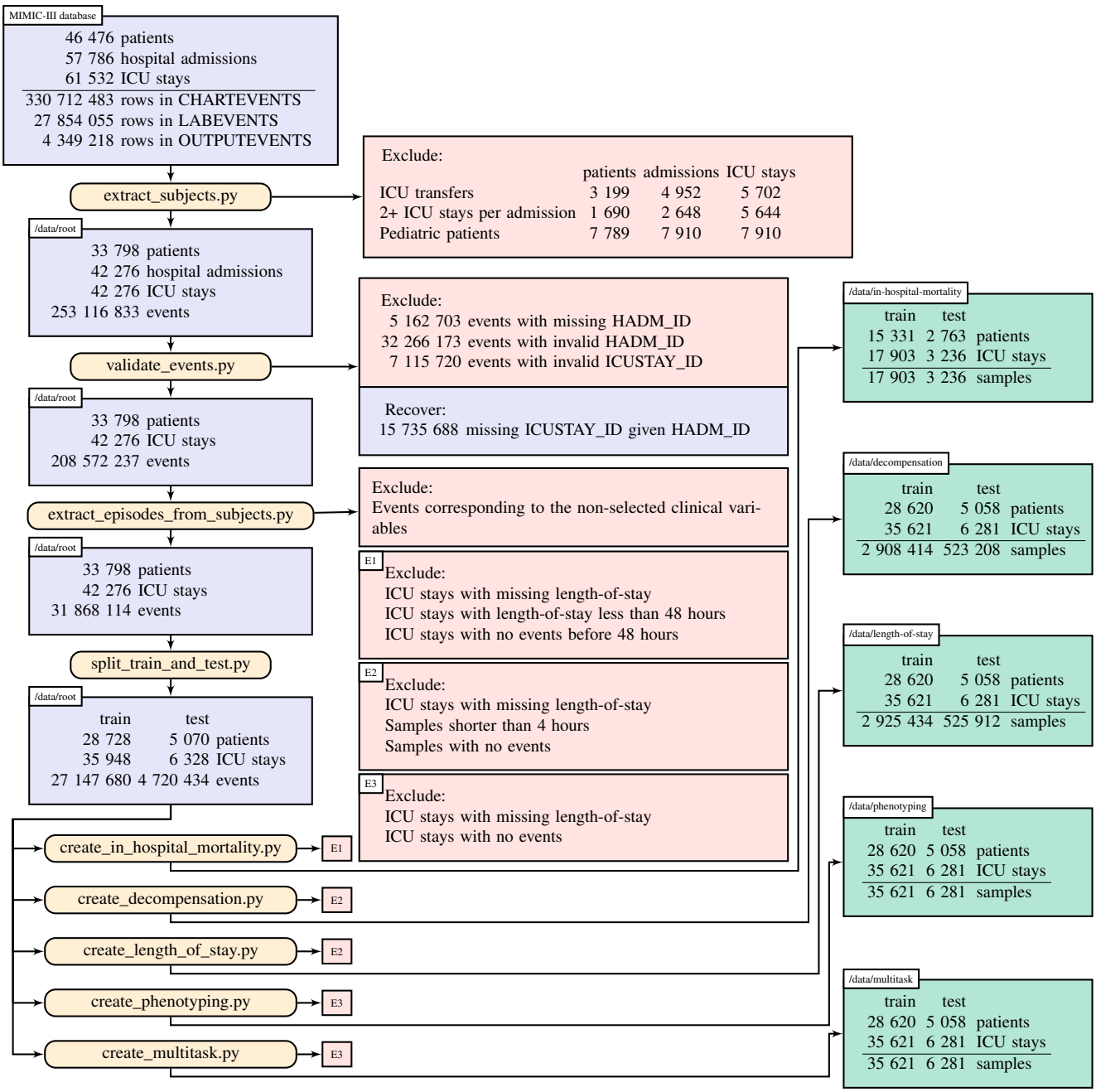


Figure 7. Benchmark generation process.

only the first 12-24 hours, but we use a wider 48-hour window to enable the detection of patterns that may indicate changes in patient acuity, similar to the PhysioNet/CinC Challenge 2012[51].

The most commonly reported metric in mortality prediction research is area under the receiver operator characteristic curve (AUC-ROC). We also report area under the precision-recall curve (AUC-PR) metric since it can be more informative when dealing with highly skewed datasets[53].

To prepare our *in-hospital-mortality* data set, we begin with the *root* cohort and further exclude all ICU stays for which LOS is unknown or less than 48 hours or for which there are no observations in the first 48 hours. This yields final training and test sets of 17,903 and 3,236 ICU stays, respectively. We determined in-hospital mortality by comparing patient date of death with hospital admission and discharge times. The resulting mortality rate is 13.23% (2,797 of 21,139 ICU stays).

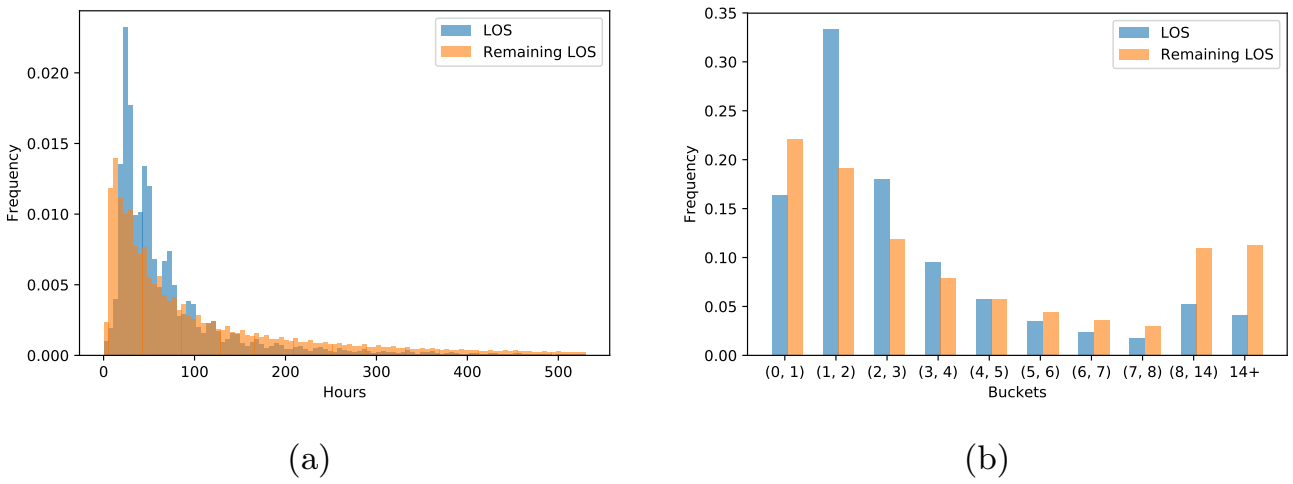


Figure 8. Distribution of length of stay (LOS). (a) The distribution of LOS for full ICU stays and remaining LOS per hour. The rightmost 5% of both distributions are not shown to keep the plot informative. (b) Histogram of bucketed patient and hourly remaining LOS (less than one day, one each for 1-7 days, between 7 and 14 days, and over 14 days).

Physiologic Decompensation

Our second benchmark task involves the detection of patients who are physiologically decompensating, or whose conditions are deteriorating rapidly. Such patients are the focus of "track-and-trigger" initiatives[3]. In such programs, patients with abnormal physiology trigger an alert, summoning a rapid response from a team of specialists who assume care of the triggering patient.

These programs are typically implemented using early warning scores, which summarize patient state with a composite score and trigger alerts based on abnormally low values. Examples include the Modified Early Warning Score (MEWS)[54], the VitalPAC Early Warning Score (ViEWS)[55], and the National Early Warning Score (NEWS)[5] being deployed throughout the United Kingdom. Like risk scores, most early warning scores are designed to be computed manually and so are based on simple thresholds and a small number of common vital signs.

Detection of decompensation is closely related to problems like condition monitoring[56] and sepsis detection[57] that have received significant attention from the machine learning community. In contrast, decompensation has seen relatively little research, with one notable exception, where Gaussian process was used to impute missing values, enabling the continuous application of early warning scores even when vitals are not recorded[58].

There are many ways to define decompensation, but most objective evaluations of early warning scores are based on accurate prediction of mortality within a fixed time window, e.g., 24 hours, after assessment[5]. Following suit, we formulate our decompensation benchmark task as a binary classification problem, in which the target label indicates whether the patient dies within the next 24 hours.

To prepare the *root* cohort for decompensation detection, we define a binary label that indicates whether the patient's date of death falls within the next 24 hours of the current time point. We then assign these labels to each hour, starting at four hours after admission to the ICU (in order to avoid having too short samples) and ending when the patient dies or is discharged. This yields 2,908,414 and 523,208 instances (individual time points with a label) in the training and test sets, respectively. The decompensation rate is 2.06% (70,696 out of 3.431,622 instances).

We use the same metrics for decompensation as for mortality, i.e., AUC-ROC and AUC-PR. Because we care about per-instance (vs. per-patient) accuracy in this task, overall performance is computed as the micro-average over all predictions, regardless of patient.

Forecasting length of stay

Our third benchmark task involves forecasting hospital LOS, one of the most important drivers of overall hospital cost[6,59]. Hospitals use patient LOS as both a measure of a patient's acuity and for scheduling and resource management[59]. Patients with extended LOS utilize more hospital resources and often have complex, persistent conditions that may not be immediately life threatening but are nonetheless difficult to treat. Reducing health care spending requires early identification and treatment of such patients.

Most LOS research has focused on identifying factors that influence LOS[60] rather than predicting it. Both severity of illness scores[61] and early warning scores[62] have been used to predict LOS but with mixed success. There has been limited machine learning research concerned with LOS, most of it focused on specific conditions[63] and cohorts[26].

LOS is naturally formulated as a regression task. Traditional research focuses on accurate prediction of LOS early in admission, but in our benchmark we predict the *remaining* LOS once per hour for every hour after admission, similar to decompensation. Such a model can be used to help hospitals and care units make decisions about staffing and resources on a regular basis, e.g., at the beginning of each day or at shift changes.

We prepare the *root* cohort for LOS forecasting in a manner similar to decompensation: for each time point, we assign a LOS target to each time point in sliding window fashion, beginning four hours after admission to the ICU and ending when the patient dies or is discharged. We compute remaining LOS by subtracting total time elapsed from the existing LOS field in MIMIC-III.

After filtering, there remain 2,925,434 and 525,912 instances (individual time points) in the training and test sets, respectively. Figure 8 (a) shows the distributions of patient LOS and hourly remaining LOS in our final cohort. Because there is no widely accepted evaluation metric for LOS predictions we use a standard regression metric – mean absolute difference (MAD).

In practice, hospitals round to the nearest day when billing, and stays over 1-2 weeks are considered extreme outliers which, if predicted, would trigger special interventions[6]. Thus, we also design a custom metric that captures how LOS is measured and studied in practice. First, we divide the range of values into ten buckets, one bucket for extremely short visits (less than one day), seven day-long buckets for each day of the first week, and two "outlier" buckets – one for stays of over one week but less than two, and one for stays of over two weeks. This converts length-of-stay prediction into an ordinal multiclass classification problem. To evaluate prediction accuracy for this problem formulation, we use Cohen's linear weighted kappa[64,65], which measures correlation between ordered items. Figure 8 (b) shows the distribution of bucketed LOS and hourly remaining LOS.

Acute care phenotype classification

Our final benchmark task is phenotyping, i.e., classifying which acute care conditions are present in a given patient record. Phenotyping has applications in cohort construction for clinical studies, comorbidity detection and risk adjustment, quality improvement and surveillance, and diagnosis[66]. Traditional research phenotypes are identified via chart review based on criteria predefined by experts, while surveillance phenotypes use simple definitions based primarily on billing, e.g., ICD-9, codes. The adoption of EHRs has led to increased interest in machine learning approaches to phenotyping that treat it as classification[67,68] or clustering[52,69]

In this task we classify 25 conditions that are common in adult ICUs, including 12 critical (and sometimes life-threatening) conditions, such as respiratory failure and sepsis; eight chronic conditions that are common comorbidities and risk factors in critical care, such as diabetes and metabolic disorders; and five conditions considered "mixed" because they are recurring or chronic with periodic acute episodes. To identify these conditions, we use the single-level definitions from the Health Cost and Utilization (HCUP) Clinical Classification Software (CCS)[70]. These definitions group ICD-9 billing and diagnostic codes into mutually exclusive, largely homogeneous disease categories, reducing some of the noise, redundancy, and ambiguity in the original ICD-9 codes. HCUP CCS code groups are used for reporting to state and national agencies, so they constitute sensible phenotype labels.

We determined phenotype labels based on the MIMIC-III ICD-9 diagnosis table. First, we mapped each code to its HCUP CCS category, retaining only the 25 categories from Table 2. We then matched diagnoses to ICU stays using the hospital admission identifier, since ICD-9 codes in MIMIC-III are associated with hospital visits, not ICU stays. By excluding hospital admissions with multiple ICU stays, we reduced some of the ambiguity in these labels: there is only one ICU stay per hospital admission with which the diagnosis can be associated. Note that we perform "retrospective" phenotype classification, in which we observe a full ICU stay before predicting which diseases are present. This is due in part to a limitation of MIMIC-III: the source of our disease labels, ICD-9 codes, do not have timestamps, so we do not know with certainty when the patient was diagnosed or first became symptomatic. Rather than attempt to assign timestamps using a heuristic, we decided instead to embrace this limitation. We apply no additional filtering to the phenotyping cohort, so there are 35,621 and 6,281 ICU stays in the training and test sets, respectively. The full list of phenotypes is shown in Table 2, along with prevalence within the benchmark data set.

Because diseases can co-occur (in fact, 99% of patients in our benchmark data set have more than one diagnosis), we formulate phenotyping as a multi-label classification problem. Similar to Lipton et al.[35], we report macro- and micro-averaged AUC-ROC with the macro-averaged score being the main score.

Baselines

In this subsection, we discuss two sets of models that we evaluate on each of our four benchmark tasks: linear (logistic) regression with hand-engineered features and LSTM-based neural networks. Both have been shown to be effective for clinical prediction from physiologic time series. For linear models, we briefly describe our feature engineering, which is also implemented in our benchmark code. For LSTMs, we review the basic definition of the LSTM architecture, our data preprocessing, and the loss function for each task. We then describe an atypical channel-wise variant of the LSTM that processes each variable separately, a deep supervision training strategy, and finally our heterogeneous multitask architecture.

Logistic regression

For our logistic regression baselines, we use a more elaborate version of the hand-engineered features described in Lipton et al.[35]. For each variable, we compute six different sample statistic features on seven different subsequences of a given time series. The per-subsequence features include minimum, maximum, mean, standard deviation, skew and number of measurements. The seven subsequences include the full time series, the first 10% of time, first 25% of time, first 50% of time, last 50% of time, last 25% of time, last 10% of time. When a sub-sequence does not contain a measurement, the features corresponding to that sub-sequence are marked missing. Notably, all categorical variables have numerical values with a meaningful scale. Thus, we use no additional encoding of categorical variables. In total, we obtain $17 \times 7 \times 6 = 714$ features per time series. The missing values are replaced are with mean values computed on the training set. Then the features are standardized by subtracting the mean and dividing by the standard deviation. We train a separate logistic regression classifier for each of mortality, decompensation, and the 25 phenotypes. For LOS, we trained a softmax regression model to solve the 10-class bucketed LOS problem.

LSTM-based models

We begin by briefly revisiting the fundamentals of long short-term memory recurrent neural networks (LSTM RNNs)[71] and introducing notation for benchmark prediction tasks in order to describe our LSTM-based models. The LSTM is a type of RNN designed to capture long term dependencies in sequential data. It takes a sequence $\{x_t\}_{t\geq 1}^{T}$ of length T as its input and outputs a T-long sequence of $\{h_t\}_{t\geq 1}^{T}$ hidden state vectors using the following equations:

$$
\begin{aligned}
i_t &= \sigma(x_t W^{(xi)} + h_{t-1} W^{(hi)}),\\
f_t &= \sigma(x_t W^{(xf)} + h_{t-1} W^{(hf)}),\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(x_t W^{(xc)} + h_{t-1} W^{(hc)} + b^{(c)}),\\
o_t &= \sigma(x_t W^{(xo)} + h_{t-1} W^{(ho)} + b^{(o)}),\\
h_t &= o_t \odot \sigma_h(c_t),
\end{aligned}
$$

where $h_0 = 0$. The σ (sigmoid) and tanh functions are applied element-wise. We do not use peephole connections[72]. The W matrices and b vectors are the trainable parameters of the LSTM. Later we will use $h_t = LSTM(x_t, h_{t-1})$ as a shorthand for the above equations. We apply dropout on non-recurrent connections between LSTM layers and before outputs.

For LSTM-based models we re-sample the time series into regularly spaced intervals. If there are multiple measurements of the same variable in the same interval, we use the value of the last measurement. We impute the missing values using the most recent measurement value if it exists and a pre-specified "normal" value otherwise (see the third column of Table 3). In addition, we also provide a binary mask input for each variable indicating the timesteps that contain a true (vs. imputed) measurement[73]. Categorical variables (even binary ones) are encoded using a one-hot vector. Numeric inputs are standardized by subtracting the mean and dividing by the standard deviation. The statistics are calculated per variable after imputation of missing values.

After the discretization and standardization steps we get 17 pairs of time series for each ICU stay: $(\{\mu_t^{(i)}\}_{t\geq 1}^{T}, \{c_t^{(i)}\}_{t\geq 1}^{T})$, where $\mu_t^{(i)}$ is a binary variable indicating whether variable i was observed at time step t and $c_t^{(i)}$ is the value (observed or imputed) of variable i at time step t. By $\{x_t\}_{t\geq 1}^{T}$ we denote the concatenation of all $\{\mu_t^{(i)}\}_{t\geq 1}^{T}$ and $\{c_t^{(i)}\}_{t\geq 1}^{T}$ time series, where concatenation is done across the axis of variables. In all our experiments x_t becomes a vector of length 76.

We also have a set of targets for each stay: $\{d_t\}_{t\geq 1}^{T}$ where $d_t \in \{0,1\}$ is a list of T binary labels for decompensation, one for each hour; $m \in \{0,1\}$ is single binary label indicating whether the patient died in-hospital; $\{\ell_t\}_{t\geq 1}^{T}$ where $\ell_t \in \mathbb{R}$ is a list of real valued numbers indicating *remaining* length of stay (hours until discharge) at each time step; and $p_{1:K} \in \{0,1\}^K$ is a vector of K binary phenotype labels. When training our models to predict length of stay, we instead use a set of categorical labels $\{l_t\}_{t\geq 1}^{T}$ where $l_t \in \{1,\ldots,10\}$ indicates in which of the ten length-of-stay buckets ℓ_t belongs. When used in the context of equations (e.g., as the output of a softmax or in a loss function), we will interpret l_t as a one-of-ten hot binary vector, indexing the ith entry as l_{ti}.

Note that because of task-specific filters are applied in the creation of benchmark tasks, we may have situations where for a given stay m is missing and/or d_t, ℓ_t are missing for some time steps. Without abusing the notation in our equations we will assume that all targets are present. In the code missing targets are discarded.

We describe the notations of the instances for each benchmark task. Each instance of in-hospital mortality prediction task is a pair $(\{x_t\}_{t\geq 1}^{48}, m)$, where x is the matrix of clinical observations of first 48 hours of the ICU stay and m is the label. An instance of decompensation and length of stay prediction tasks is a pair $(\{x_t\}_{t\geq 1}^{\tau}, y)$, where x is the matrix of clinical observations of first τ hours of the stay and y is the target variable (either d_τ, ℓ_τ or l_τ). Each instance of phenotype classification task is a pair $(\{x_t\}_{t\geq 1}^{T}, p_{1:K})$, where x is the matrix of observation of the whole ICU stay and $p_{1:K}$ are the phenotype labels.

Our first LSTM-based baseline takes an instance $(\{x_t\}_{t\geq 1}^T, y)$ of a prediction task and uses a single LSTM layer to process the input: $h_t = LSTM(x_t, h_{t-1})$. To predict the target we add the output layer:

$$\begin{aligned}
\widehat{d_T} &= \sigma\left(w^{(d)}h_T + b^{(d)}\right),\\
\widehat{m} &= \sigma\left(w^{(m)}h_T + b^{(m)}\right),\\
\widehat{\ell_T} &= \text{relu}\left(w^{(\ell)}h_T + b^{(\ell)}\right),\\
\widehat{l_T} &= \text{softmax}\left(W^{(l)}h_T + b^{(l)}\right),\\
\widehat{p_i} &= \sigma\left(W^{(p)}_{i,\cdot}h_T + b^{(p)}_i\right),
\end{aligned}$$

where y is d_T, m, ℓ_T, l_T or $p_{1:K}$ respectively. The loss functions we use to train these models are (in the same order as above):

$$\begin{aligned}
\mathscr{L}_d &= CE(d_T, \widehat{d_T}),\\
\mathscr{L}_m &= CE(m, \widehat{m}),\\
\mathscr{L}_\ell &= (\widehat{\ell_T} - \ell_T)^2,\\
\mathscr{L}_l &= MCE(l_T, \widehat{l_T}),\\
\mathscr{L}_p &= \frac{1}{K}\sum_{k=1}^{K} CE(p_k, \widehat{p_k}),
\end{aligned}$$

where $CE(y,\widehat{y})$ is the binary cross entropy and $MCE(y,\widehat{y})$ is multiclass cross entropy defined over the C classes:

$$\begin{aligned}
\text{CE}(y,\widehat{y}) &= -\left(y\cdot\log(\widehat{y}) + (1-y)\cdot\log(1-\widehat{y})\right),\\
\text{MCE}(y,\widehat{y}) &= -\sum_{k=1}^{C} y_k \log(\widehat{y}_k).
\end{aligned}$$

We call this model "Standard LSTM".

Channel-wise LSTM

In addition to the standard LSTM baseline, we also propose a modified LSTM baseline which we call channel-wise LSTM. While the standard LSTM network work directly on the concatenation $\{x_t\}_{t\geq 1}^T$ of the time series, the channel-wise LSTM pre-processes the data $(\{\mu_t^{(i)}\}_{t\geq 1}^T, \{c_t^{(i)}\}_{t\geq 1}^T)$ of different variables independently using a bidirectional LSTM layer. We use different LSTM layers for different variables. Then the outputs of these LSTM layers are concatenated and are fed to another LSTM layer.

$$\begin{aligned}
p_t^{(i)} &= LSTM([\mu_t^{(i)}; c_t^{(i)}], p_{t-1}^{(i)})\\
q_t^{(i)} &= LSTM([\overleftarrow{\mu^{(i)}}_t; \overleftarrow{c^{(i)}}_t], q_{t-1}^{(i)})\\
u_t &= [p_t^{(1)}; \overleftarrow{q^{(1)}}_t; \ldots p_t^{(17)}; \overleftarrow{q^{(17)}}_t]\\
h_t &= LSTM(u_t, h_{t-1})
\end{aligned}$$

$\overleftarrow{x}_t$ denotes the t-th element of the reverse of the sequence $\{x_t\}_{t\geq 1}^T$. The output layers and loss functions for each task are the same as those in the standard LSTM baseline.

The intuition behind having channel-wise module is two-fold. First, it helps to pre-process the data of a single variable before mixing it with the data of other variables. This way the model can learn to store some useful information related to only that particular variable. For example, it can learn to store the maximum heart rate or the average blood pressure in earlier time steps. This kind of information is hard to learn in the case of standard LSTMs, as the input to hidden weight matrices need to have sparse rows. Second, this channel-wise module facilitates incorporation of missing data information by explicitly showing which mask variables relate to which variables. This information can be tricky to learn in standard LSTM models. While in most of our experiments channel-wise LSTMs outperformed standard LSTMs, we did not perform an extensive ablation study to determine the contribution of various components of channel-wise LSTMs (grouping of related variables, bidirectionality, the additional LSTM layer) in the performance gain.

Note that this channel-wise module can be used as a replacement of the input layer in any neural architecture which takes the concatenation of time series of different variables as its input.

Deep supervision

So far we defined models that do the prediction in the last step. This way the supervision comes from the last time step, implying that the model needs to pass information across many time steps. We propose two methods where we supervise the model at each time step. We use the term deep supervision to refer them.

For in-hospital mortality and phenotype prediction tasks we use target replication[35] to do deep supervision. In this approach we replicate the target in all time steps and by changing the loss function we require the model to predict the replicated target variable too. The loss functions of these deeply supervised models become:

$$\mathscr{L}_m^* = (1-\alpha) * \mathrm{CE}(m, \widehat{m_T}) + \alpha * \frac{1}{T}\sum_{t=1}^{T} \mathrm{CE}(m, \widehat{m_t}),$$

$$\mathscr{L}_p^* = \frac{1}{K}\sum_{i=1}^{K}\left((1-\alpha) * \mathrm{CE}(p_k, \widehat{p}_{T,k}) + \alpha * \frac{1}{T}\sum_{t=1}^{T} \mathrm{CE}(p_k, \widehat{p}_{tk})\right),$$

where $\alpha \in [0,1]$ is a hyperparameter that represents the strength of target replication part in loss functions, $\widehat{d_t}$ is decompensation prediction at time step t, and $\widehat{p_{tk}}$ is the prediction of k-th phenotype a time step t.

For decompensation and length of stay prediction tasks we cannot use target replication, because the target of the last time step might be wrong for the other time steps. Since in these tasks we create multiple prediction instances from a single ICU stay, we can group these samples and predict them in a single pass. This way we will have targets for each time step and the model will be supervised at each time step. The loss functions of these deeply supervised models are:

$$\mathscr{L}_d^* = \frac{1}{T}\sum_{t=1}^{T} \mathrm{CE}(d_t, \widehat{d_t}),$$

$$\mathscr{L}_\ell^* = \frac{1}{T}\sum_{t=1}^{T} (\widehat{\ell_t} - \ell_t)^2,$$

$$\mathscr{L}_l^* = \frac{1}{T}\sum_{t=1}^{T} \mathrm{MCE}(l_t, \widehat{l_t}).$$

Note that whenever we group the instances of a single ICU stay, we use simple left-to-right LSTMs instead of bidirectional LSTMs, so that the data from future time steps is not used.

Multitask learning LSTM

So far we predicted targets for each task independently. There is a natural question about the effectiveness of multitasking. Correlations between the targets of different tasks are presented in the Figure 9. For each task we propose another baseline, where we try to use the other three tasks as auxiliary tasks to enhance the performance. This multitasking can be done with either standard LSTM or channel-wise LSTM.

In multitask setting we group the instances coming from a single ICU stay and predict all targets associated with a single ICU jointly. This means we use deep supervision of decompensation and length of stay prediction tasks. We are free to choose whether we want to use deep supervision for in-hospital mortality and phenotype prediction tasks.

For in-hospital mortality, we consider only the first 48 timesteps $\{x_t\}_{t\geq 1}^{t_m}$, and predict $\widehat{m}$ at $t_m = 48$ by adding a single dense layer with sigmoid activation which takes h_{t_m} as its input. For decompensation, we take the full sequence $\{x_t\}_{t\geq 1}^{T}$ and generate a sequence of mortality predictions $\{\widehat{d}\}_{t\geq 1}^{T}$ by adding a dense layer at every step. For phenotyping, we consider the full sequence but predict phenotypes $\widehat{p}$ only at the last timestep T by adding 25 parallel dense layers with sigmoid activations. Similar to decompensation, we predict LOS by adding a single dense layer at each timestep. We experiment with two settings. In one case the dense layer outputs a single real number $\widehat{\ell_t}$, in the other case it uses softmax activation to output a distribution over the ten LOS buckets $\widehat{l_t}$. The full multitask LSTM architecture is illustrated in Figure 10.

The loss functions for each task are the same as those in deep supervised setting. The overall loss is a weighted sum of task-specific losses:

$$\mathscr{L}_{mt} = \lambda_d \cdot \mathscr{L}_d^* + \lambda_l \cdot \mathscr{L}_l^* + \lambda_m \cdot \mathscr{L}_m^* + \lambda_p \cdot \mathscr{L}_p^*,$$

where the weights are non-negative numbers. For raw length of stay prediction we replace $\mathscr{L}_l^*$ with $\mathscr{L}_\ell^*$ in the multitasking loss function.

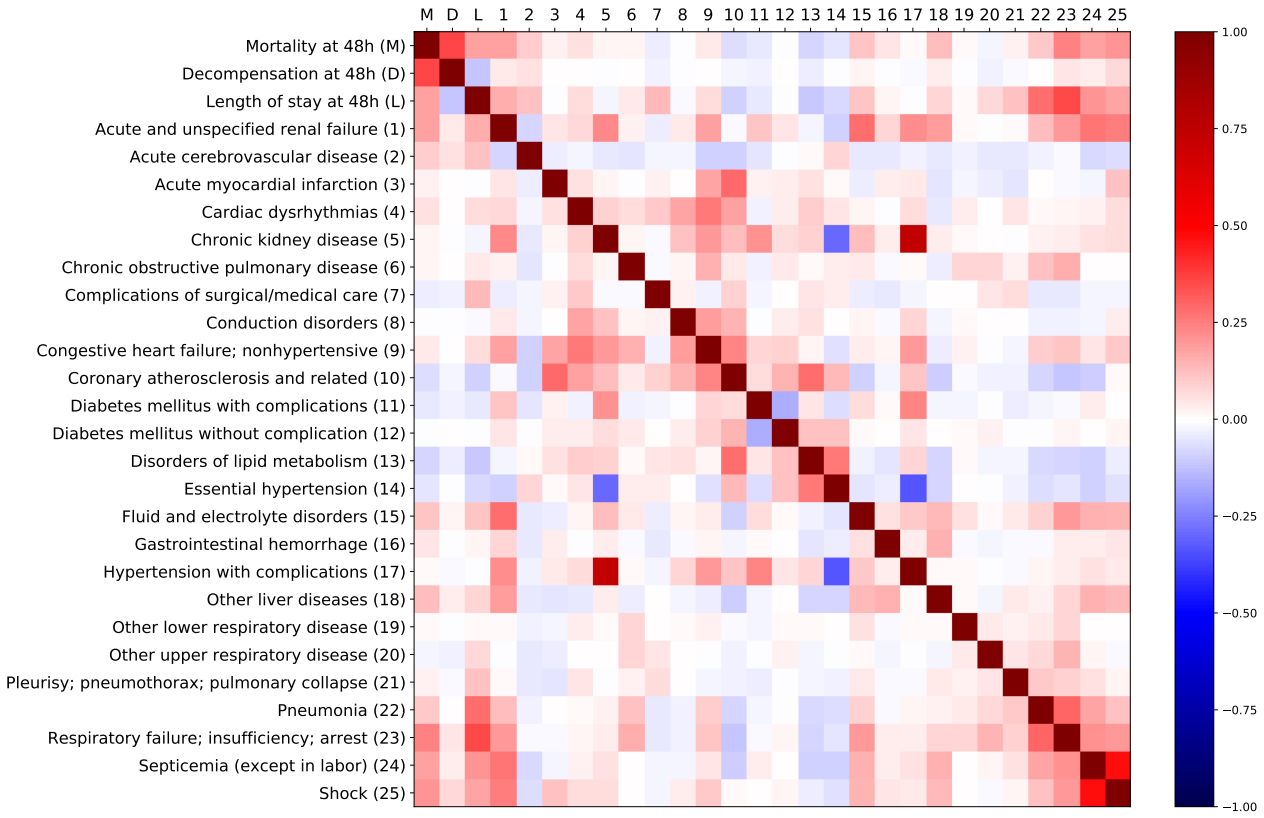


Figure 9. Correlations between task labels.

Experiments, Model selection and Evaluation

For all algorithms we use the data of the same 15% patients of the predefined training set as validation data and train the models on the remaining 85%. We use grid search to tune all hyperparameters based on validation set performance. The best model for each baseline is chosen according to the performance on the validation set. The final scores are reported on the test set, which we used sparingly during model development in order to avoid unintentional test set leakage.

The only hyperparameters of logistic regression models are the coefficients of L1 and L2 regularizers. In our experiments L1 and L2 penalties are mutually exclusive and are not applied to bias terms. We used grid search to find the best penalty type and its regularization coefficient. For L2 penalty the grid of inverse of regularization coefficient includes $10^{-5}, 10^{-4}, \ldots, 1$, while for L1 penalty it includes $10^{-4}, 10^{-3}, \ldots, 1$. For in-hospital mortality and decompensation prediction, the best performing logistic regression used L2 regularization with $C = 0.001$. For phenotype prediction, the best performing logistic regression used L1 regularization with $C = 0.1$. For LOS prediction, the best performing logistic regression used L2 regularization with $C = 10^{-5}$.

When discretizing the data for LSTM-based models, we set the length of regularly spaced intervals to 1 hour. This gives a reasonable balance between amount of missing data and number of measurements of the same variable that fall into the same interval. This choice also agrees with the rate of sampling prediction instances for decompensation and LOS prediction tasks. We also tried to use intervals of length 0.8 hours, but there was no improvement in the results. For LSTM-based models, hyperparameters include the number of memory cells in LSTM layers, the dropout rate, and whether to use one or two LSTM layers. Channel-wise LSTM models have one more hyperparameter - the number of units in channel-wise LSTMs (all the 17 LSTMs having the same number of units). Whenever the target replication is enabled, we set $\alpha = 0.5$ in the corresponding loss function.

The best values of hyperparameters of LSTM-based models vary across the tasks. They are listed in `pretrained_models.md` file in our code repository[78]. Generally, we noticed that dropout helps a lot to reduce overfitting. In fact, all LSTM-based baselines for in-hospital mortality prediction task (where the problem of overfitting is the most severe) use 30% dropout.

For multitask models we have 4 more hyperparameters: λ_d, λ_m, λ_l and λ_p weights in the loss function. We didn't do full grid search for tuning these hyperparameters. Instead we tried 5 different values of $(\lambda_d, \lambda_m, \lambda_l, \lambda_p)$: $(1,1,1,1)$; $(4,2.5,0.3,1)$; $(1,0.4,3,1)$; $(1,0.2,1.5,1)$ and $(0.1,0.1,0.5,1)$. The first has the same weighs for each task, while the second tries to make the

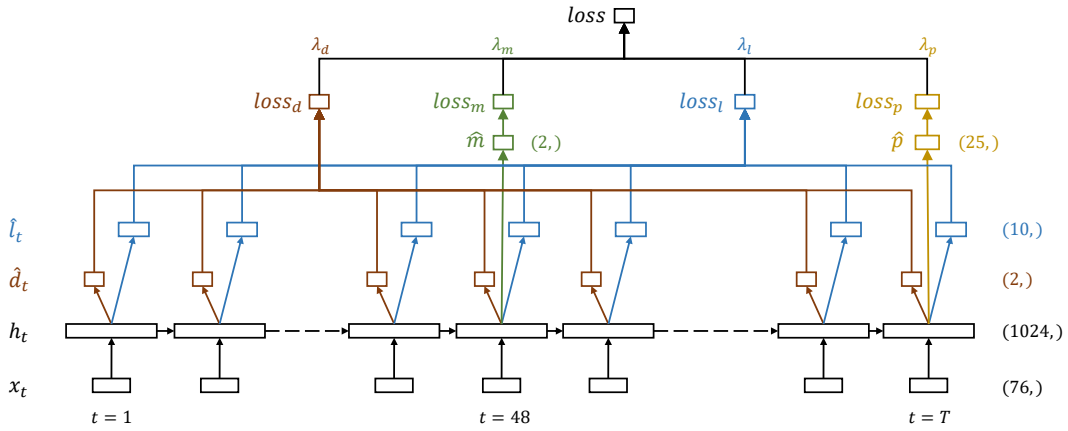


Figure 10. LSTM-based network architecture for multitask learning.

four summands of the loss function approximately equal. The three remaining combinations were selected by looking at the speeds of learning of each task.

Overfitting was a serious problem in multitasking setup, with mortality and decompensation prediction validation performance degrading faster than the others. All of the best multitask baselines use either $(1, 0.2, 1.5, 1)$ or $(0.1, 0.1, 0.5, 1)$ for $(\lambda_d, \lambda_m, \lambda_l, \lambda_p)$. The first configuration performed the best for in-hospital mortality, decompensation and length of stay prediction tasks, whereas the second configuration was better for phenotype prediction task. The fact that λ_d, λ_m and λ_l of the best multitask baselines for phenotype prediction task are relatively small supports the hypothesis that additional multitasking in phenotype prediction task hurts the performance.

All LSTM-based models are trained using ADAM[74] with a 10^{-3} learning rate and $\beta_1 = 0.9$. The batch size is set to either 8, 16 or 32 depending on the memory available at the computational unit. We did not do extensive experiments to check whether tuning the hyperparameters of the optimizer and the batch size improves performance.

Finally, we evaluate the best baselines on their corresponding test sets. Since the test score is an estimate of model performance on unseen examples, we use bootstrapping to estimate confidence intervals of the score. Bootstrapping have been used to estimate the standard deviations of the evaluation measures[75], to compute statistically significant differences between different models[76] and to report 95% bootstrap confidence intervals for the models[25,77]. Providing confidence intervals also helps us to fight against a known problem of all public benchmark datasets – overfitting on the test set. To estimate a 95% confidence interval we resample the test set K times; calculate the score on the resampled sets; and use 2.5 and 97.5 percentiles of these scores as our confidence interval estimate. For in-hospital mortality and phenotype prediction K is 10000, while for decompensation and length-of-stay prediction K is 1000, since the test sets of these tasks are much bigger.

Code availability

The code to reproduce the results is available on Zenodo repository[78].

Data availability

MIMIC-III database analyzed in this study is available on PhysioNet repository[21]. The code to generate the datasets used in this study is available on Zenodo repository[78].

Acknowledgements

The authors would like to thank Zachary Lipton for helpful comments. H.H. and H.K. were partially funded by an ISTC research grant. G.V.S. and A.G. acknowledge support by the Defense Advanced Research Projects Agency (DARPA) under grant FA8750-17-C-0106.

Author contributions

D.K. had the initial idea for the analysis. D.K., G.V.S. and A.G. designed the study. H.H. and D.K. designed the data processing pipeline. H.H. and H.K. designed machine learning experiments. H.H. performed the experiments. H.H. H.K. and D.K. analyzed the results and drafted the paper. H.K., G.V.S. and A.G. revised the paper draft.

Competing interests

The authors declare no competing interests.

References

1. *Introduction to the HCUP National Inpatient Sample (NIS) 2012*. (Agency for Healthcare Research and Quality, 2014).
2. Henry, J., Pylypchuk, Y., Talisha Searcy, M. & Patel, V. Adoption of electronic health record systems among US non-federal acute care hospitals: 2008-2015. *ONC Data Brief* **35** (Office of the National Coordinator for Health Information Technology, Washington DC, USA, 2015).
3. Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A. & Escobar, G. Big data in health care: using analytics to identify and manage high-risk and high-cost patients. *Heal. Aff.* **33**, 1123–1131 (2014).
4. Zimmerman, J. E., Kramer, A. A., McNair, D. S. & Malila, F. M. Acute physiology and chronic health evaluation (apache) iv: hospital mortality assessment for today's critically ill patients. *Crit. Care Med.* **34**, 1297–1310 (2006).
5. Williams, B. *et al. National Early Warning Score (NEWS): Standardising the assessment of acute-illness severity in the NHS*. (London: The Royal College of Physicians, 2012).
6. Dahl, D. *et al.* The high cost of low-acuity icu outliers. *J. Healthc. Manag.* **57**, 421–433 (2012).
7. Saria, S. & Goldenberg, A. Subtyping: What it is and its role in precision medicine. *IEEE Intell. Syst.* **30**, 70–75 (2015).
8. Iserson, K. V. & Moskop, J. C. Triage in medicine, part i: concept, history, and types. *Ann. Emerg. Med.* **49**, 275–281 (2007).
9. Apgar, V. A proposal for a new method of evaluation of the newborn. *Curr. Res. Anesth. Analg.* **32**, 260–267 (1952).
10. Ferrucci, D., Levas, A., Bagchi, S., Gondek, D. & Mueller, E. T. Watson: beyond jeopardy! *Artif. Intell.* **199**, 93–105 (2013).
11. Silver, D. *et al.* Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
12. Caballero Barajas, K. L. & Akella, R. Dynamically modeling patient's health state from electronic medical records: A time series approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 69–78 (ACM, Sydney, Australia, 2015).
13. Ghassemi, M. *et al.* A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 446–453 (AAAI Press, Austin, Texas, 2015).
14. Luo, Y., Xin, Y., Joshi, R., Celi, L. & Szolovits, P. Predicting icu mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 42–50 (AAAI Press, Phoenix, Arizona, 2016).
15. Lee, J. & Maslove, D. M. Customization of a severity of illness score using local electronic medical record data. *J. intensive care medicine* **32**, 38–47 (2017).
16. Johnson, A., Pollard, T. & Mark, R. Reproducibility in critical care: a mortality prediction case study. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, vol. 68, 361–376 (PMLR, Boston, Massachusetts, 2017).
17. Quinn, J. A., Williams, C. K. & McIntosh, N. Factorial switching linear dynamical systems applied to physiological condition monitoring. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 1537–1551 (2009).
18. Laxmisan, A. *et al.* The multitasking clinician: Decision-making and cognitive demand during and after team handoffs in emergency care. *Int. J. Med. Inform.* **76**, 801 – 811 (2007).
19. Horn, S. D. *et al.* The relationship between severity of illness and hospital length of stay and mortality. *Med. Care* **29**, 305–317 (1991).
20. Johnson, A. E. W. *et al.* MIMIC-III, a freely accessible critical care database. *Sci. Data* **3**, 160035 (2016).
21. Laboratory For Computational Physiology, M. I. T. The MIMIC-III clinical database. *PhysioNet* https://doi.org/10.13026/C2XW26 (2015).
22. Caruana, R., Baluja, S. & Mitchell, T. Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in Neural Information Processing Systems 8*, 959–965 (MIT Press, Denver, Colorado, 1996).

23. Clermont, G., Angus, D. C., DiRusso, S. M., Griffin, M. & Linde-Zwirble, W. T. Predicting hospital mortality for patients in the intensive care unit: a comparison of artificial neural networks with logistic fmultion models. *Crit. Care Med.* **29**, 291–296 (2001).

24. Celi, L. A. *et al.* A database-driven decision support system: customized mortality prediction. *J. Pers. Med.* **2**, 138–148 (2012).

25. Rajkomar, A. *et al.* Scalable and accurate deep learning with electronic health records. *NPJ Digit. Med.* **1**, 18 (2018).

26. Grigsby, J., Kooken, R. & Hershberger, J. Simulated neural networks to predict outcomes, costs, and length of stay among orthopedic rehabilitation patients. *Arch. Phys. Med. Rehabil.* **75**, 1077–1081 (1994).

27. Mobley, B. A., Leasure, R. & Davidson, L. Artificial neural network predictions of lengths of stay on a post-coronary care unit. *Hear. & Lung: The J. Acute Critical Care* **24**, 251–256 (1995).

28. Buchman, T. G., Kubos, K. L., Seidler, A. J. & Siegforth, M. J. A comparison of statistical and connectionist models for the prediction of chronicity in a surgical intensive care unit. *Crit. Care Med.* **22**, 750–762 (1994).

29. Yousefi, S., Song, C., Nauata, N. & Cooper, L. Learning genomic representations to predict clinical outcomes in cancer. Preprint at https://arxiv.org/abs/1609.08663 (2016).

30. Ranganath, R., Perotte, A., Elhadad, N. & Blei, D. Deep survival analysis. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, vol. 56 (PMLR, Los Angeles, California, USA, 2016).

31. Lasko, T. A., Denny, J. C. & Levy, M. A. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PLoS ONE* **8**, e66341 (2013).

32. Che, Z., Kale, D., Li, W., Bahadori, M. T. & Liu, Y. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 507–516 (ACM, Sydney, Australia, 2015).

33. Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F. & Sun, J. Doctor AI: Predicting clinical events via recurrent neural networks. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, vol. 56 (PMLR, Los Angeles, California, USA, 2016).

34. Razavian, N., Marcus, J. & Sontag, D. Multi-task prediction of disease onsets from longitudinal lab tests. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, vol. 56 (PMLR, Los Angeles, California, USA, 2016).

35. Lipton, Z. C., Kale, D. C., Elkan, C. & Wetzel, R. Learning to diagnose with LSTM recurrent neural networks. In *International Conference on Learning Representations* (San Juan, Puerto Rico, 2016).

36. Ngufor, C., Upadhyaya, S., Murphree, D., Kor, D. & Pathak, J. Multi-task learning with selective cross-task transfer for predicting bleeding and other important patient outcomes. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 1–8 (IEEE, Paris, France, 2015).

37. Collobert, R. & Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167 (ACM, Helsinki, Finland, 2008).

38. Gupta, P., Malhotra, P., Vig, L. & Shroff, G. Using features from pre-trained timenet for clinical predictions. In *Proceedings of the 3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, 38–44 (Stockholm, Sweden, 2018).

39. Gupta, P., Malhotra, P., Vig, L. & Shroff, G. Transfer learning for clinical time series analysis using recurrent neural networks. In *Machine Learning for Medicine and Healthcare Workshop at ACM KDD 2018 Conference* (London, United Kingdom, 2018).

40. Jin, M. *et al.* Improving hospital mortality prediction with medical named entities and multimodal learning. In *Machine Learning for Health (ML4H) Workshop at NeurIPS* (Montreal, Canada, 2018).

41. Oh, J., Wang, J. & Wiens, J. Learning to exploit invariances in clinical time-series data using sequence transformer networks. In *Proceedings of the 3rd Machine Learning for Healthcare Conference*, vol. 85, 332–347 (PMLR, Palo Alto, California, USA, 2018).

42. Malone, B., Garcia-Duran, A. & Niepert, M. Learning representations of missing data for predicting patient outcomes. Preprint at https://arxiv.org/abs/1811.04752 (2018).

43. Chang, C.-H., Mai, M. & Goldenberg, A. Dynamic measurement scheduling for adverse event forecasting using deep RL. In *Machine Learning for Health (ML4H) Workshop at NeurIPS* (Montreal, Canada, 2018).

44. Xu, Y., Biswal, S., Deshpande, S. R., Maher, K. O. & Sun, J. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2565–2573 (ACM, London, United Kingdom, 2018).

45. Chung, I., Kim, S., Lee, J., Hwang, S. J. & Yang, E. Mixed effect composite RNN-GP: A personalized and reliable prediction model for healthcare. Preprint at https://arxiv.org/abs/1806.01551 (2018).

46. Bahadori, M. T. Spectral capsule networks. In *International Conference on Learning Representations Workshop Track* (New Orleans, Louisiana, USA, 2018).

47. Rafi, P., Pakbin, A. & Pentyala, S. K. Interpretable deep learning framework for predicting all-cause 30-day ICU readmissions. *Tech. Rep.*, (Texas A&M University, 2018).

48. Song, H., Rajan, D., Thiagarajan, J. J. & Spanias, A. Attend and diagnose: Clinical time series analysis using attention models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence* (AAAI Press, New Orleans, Louisiana, USA, 2018).

49. Purushotham, S., Meng, C., Che, Z. & Liu, Y. Benchmarking deep learning models on large healthcare datasets. *J. Biomed. Inform.* **83**, 112 – 134 (2018).

50. Le Gall, J.-R. *et al.* A simplified acute physiology score for icu patients. *Crit. Care Med.* **12**, 975–977 (1984).

51. Silva, I., Moody, G., Scott, D. J., Celi, L. A. & Mark, R. G. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, 245–248 (IEEE, Krakow, Poland, 2012).

52. Marlin, B. M., Kale, D. C., Khemani, R. G. & Wetzel, R. C. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, 389–398 (ACM, Miami, Florida, 2012).

53. Davis, J. & Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, 233–240 (ACM, Pittsburgh, Pennsylvania, USA, 2006).

54. Subbe, C., Kruger, M., Rutherford, P. & Gemmel, L. Validation of a modified early warning score in medical admissions. *Qjm* **94**, 521–526 (2001).

55. Prytherch, D. R., Smith, G. B., Schmidt, P. E. & Featherstone, P. I. Views – towards a national early warning score for detecting adult inpatient deterioration. *Resuscitation* **81**, 932–937 (2010).

56. Aleks, N. *et al.* Probabilistic detection of short events, with application to critical care monitoring. In *Advances in Neural Information Processing Systems 21*, 49–56 (Curran Associates, Inc., Vancouver, Canada, 2009).

57. Henry, K. E., Hager, D. N., Pronovost, P. J. & Saria, S. A targeted real-time early warning score (trewscore) for septic shock. *Sci. Transl. Med.* **7**, 299ra122–299ra122 (2015).

58. Clifton, L., Clifton, D. A., Pimentel, M. A., Watkinson, P. J. & Tarassenko, L. Gaussian process regression in vital-sign early warning systems. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 6161–6164 (IEEE, San Diego, California, USA, 2012).

59. Romano, P., Hussey P. & Ritley, D.. *Selecting quality and resource use measures: A decision guide for community quality collaboratives*. (Agency for Healthcare Research and Quality, 2014).

60. Higgins, T. L. *et al.* Early indicators of prolonged intensive care unit stay: Impact of illness severity, physician staffing, and pre–intensive care unit length of stay. *Crit. Care Med.* **31**, 45–51 (2003).

61. Osler, T. M. *et al.* Predicting survival, length of stay, and cost in the surgical intensive care unit: Apache ii versus iciss. *J. Trauma Acute Care Surg.* **45**, 234–238 (1998).

62. Paterson, R. *et al.* Prediction of in-hospital mortality and length of stay using an early warning scoring system: clinical audit. *Clin. Medicine* **6**, 281–284 (2006).

63. Pofahl, W. E., Walczak, S. M., Rhone, E. & Izenberg, S. D. Use of an artificial neural network to predict length of stay in acute pancreatitis. *The Am. Surg.* **64**, 868 (1998).

64. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**, 37–46 (1960).

65. Brennan, R. L. & Prediger, D. J. Coefficient kappa: Some uses, misuses, and alternatives. *Educ. Psychol. Meas.* **41**, 687–699 (1981).

66. Oellrich, A. *et al.* The digital revolution in phenotyping. *Brief. Bioinform.* **17**, 819–830 (2015).

67. Agarwal, V. *et al.* Learning statistical models of phenotypes using noisy labeled training data. *J. Am. Med. Informatics Assoc.* **23**, 1166 (2016).

68. Halpern, Y., Horng, S., Choi, Y. & Sontag, D. Electronic medical record phenotyping using the anchor and learn framework. *J. Am. Med. Informatics Assoc.* **23**, 731 (2016).

69. Ho, J. C., Ghosh, J. & Sun, J. Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 115–124 (ACM, New York, New York, USA, 2014).

70. *Clinical Classifications Software (CCS) for ICD-9-CM fact sheet* (Agency for Healthcare Research and Quality, 2012).

71. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neur. Comp.* **9**, 1735–1780 (1997).

72. Gers, F. A. & Schmidhuber, J. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, vol. 3, 189–194 (IEEE, Como, Italy, 2000).

73. Lipton, Z. C., Kale, D. C. & Wetzel, R. Modeling missing data in clinical time series with rnns. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, vol. 56 (PMLR, Los Angeles, California, USA, 2016).

74. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).

75. Choi, E. *et al.* Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems 29*, 3504–3512 (Curran Associates, Inc., Barcelona, Spain, 2016).

76. Smith, L. *et al.* Overview of biocreative ii gene mention recognition. *Genome Biol.* **9**, S2 (2008).

77. Rajpurkar, P. *et al.* CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. Preprint at https://arxiv.org/abs/1711.05225 (2017).

78. Harutyunyan, H. *et al.* MIMIC-III benchmark repository. *Zenodo* https://doi.org/10.5281/zenodo.1306527 (2018).