Directly Modeling Missing Data in Sequences with RNNs: Improved Classification of Clinical Time Series

Zachary C. Lipton   ZLIPTON@CS.UCSD.EDU
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093, USA

David C. Kale   DKALE@USC.EDU
USC Information Sciences Institute
Marina del Rey, CA, USA

Randall Wetzel   RWETZEL@CHLA.USC.EDU
Laura P. and Leland K. Whittier Virtual Pediatric Intensive Care Unit
Children's Hospital LA
Los Angeles, CA 90089

Abstract

We demonstrate a simple strategy to cope with missing data in sequential inputs, addressing the task of multilabel classification of diagnoses given clinical time series. Collected from the intensive care unit (ICU) of a major urban medical center, our data consists of multivariate time series of observations. The data is irregularly sampled, leading to missingness patterns in re-sampled sequences. In this work, we show the remarkable ability of RNNs to make effective use of binary indicators to directly model missing data, improving AUC and F1 significantly. However, while RNNs can learn arbitrary functions of the missing data and observations, linear models can only learn substitution values. For linear models and MLPs, we show an alternative strategy to capture this signal. Additionally, we evaluate LSTMs, MLPs, and linear models trained on missingness patterns only, showing that for several diseases, *what tests are run* can be more predictive than the results themselves.

1. Introduction

Hospital intensive care units (ICUs) record large amounts data for each admitted patient. Electrocardiograms, pulse oximeters, and arterial blood pressure sensors stream vital signs, which clinical staff chart routinely during hourly rounds. Lab test results and medications are recorded as they are ordered or delivered by physicians and nurses. Critically ill patients may receive invasive interventions, such as tracheal intubation, enabling the monitoring of inspired and expired gases. The result are rich sequences of clinical observations depicting both the patient's health and the care she receives over time. We would like to mine these time series to build accurate predictive models for diagnosis and other applications. Modern machine learning techniques are well-suited to learning sequential or temporal relationships from such time series. Recurrent neural networks (RNNs), in particular, offer unprecedented predictive power in myriad sequence learning domains, including, natural language, speech, video, handwriting, among others. Recently, Lipton et al. (2016), demonstrated the efficacy of RNNS for multilabel classification of diagnoses in clinical time series data.

However, medical time series data presents modeling problems not found in the clean academic datasets on which most RNN research focuses. Clinical observations are sampled irregularly, with rates varying between patients, across variables, and even through time. One common strategy for dealing with sparse, variable sampling is to re-sample the data to discrete time series, but the resulting sequences often contain missing values (Marlin et al., 2012) Most often, researchers fill missing values using a heuristic or unsupervised imputation strategy (Lasko et al., 2013). Adding complication, these values are not missing at random, but reflect decisions by caregivers. Thus, the pattern of recorded measurements contain potential information about the state of the patient.

In this work, we directly model missing values as features, extending the methodology of Lipton et al. (2016) for RNN-based multilabel prediction of diagnoses given multivariate time series of measurements from the intensive care unit at a major urban hospital. Unlike Lipton et al. (2016), who approach missing data via heuristic imputation, we directly model missingness as a first-class feature. We show that RNNs make remarkable use of binary indicators for missing data, improving AUC and F1 significantly. However, while RNNs can learn arbitrary functions of the missingness indicators and observations, linear models can only learn substitution values. Multilayer perceptrons (MLPs) trained on fixed windows can also make use of the information from missing data indicators, but their overall performance suffers due to their inability to model full patient trajectories. For linear models and MLPs, we show an alternative strategy to capture this signal using a small number of simple hand-engineered features. Additionally, we evaluate RNNs, MLPs, and linear models trained on missingness patterns only, showing that for several diseases, *what tests are run* can be as predictive as or more predictive than the actual measurements.

Of course, we may not want our predictive models to rely upon the patterns of treatment. as argued by Caruana et al. (2015). Once deployed, our models may influence the treatment protocols, thus invalidating their predictions. Nonetheless, doctors at present often do utilize knowledge of past care , and treatment signal can leak into the actual measurements themselves in a way that is detectable by sufficiently powerful models. As a final contribution of this paper, we present critical discussion of these practical and philosophical issues and argue that the proper way forward in our field is to directly model signals from missing values, utilization, and treatment, rather than ignore or attempt to remove them.

2. Data

Our dataset consists of patient records extracted from a large urban medical center as part of an IRB-approved study. In all, the dataset contains roughly 10,000 ICU episodes. Each episode describes the stay of one patient in the ICU for a period of at least 12 hours. In addition, each patient record contains a static set of diagnostic codes, annotated by physicians either during or after each ICU visit.

2.1 Inputs

In their rawest representation, episodes are irregularly sampled multivariate time series of 13 variables: diastolic and systolic blood pressure, peripheral capillary refill rate, end-tidal CO_2 (ETCO$_2$), fraction of inspired O_2 (FIO$_2$), total Glascow coma scale, blood glucose, heart rate, pH, respiratory rate, blood oxygen saturation, body temperature, and urine

output. To render our data suitable for learning with RNNs, we re-sample time series to sequences of hourly time steps, where time step t covers the interval between hours t and $t + 1$, closed on the left but open on the right. Because actual admission times are not recorded reliably, we use the time of the first recorded observation as time step $t = 0$. We combine multiple measurements within the same hour window by taking their mean.

Vital signs such as heart rate are measured on average just over once per hour, while lab tests requiring a blood draw (e.g., glucose) are measured just over once per day (see Appendix B for further detail). In addition, the timing of and time between observations varies across patients and over time. The resulting sequential representation will have many missing values, and other variables are missing altogether.

Note that our methods do not depend on choice of re-sampling rate but are sensitive to it. A higher re-sampling rate of, e.g., every half hour, would double the length of the sequences, making learning via backpropagation through time more challenging (Bengio et al., 1994). For our data, this cost is not justified because the most frequently measured variables (vital signs) are recorded about once per hour. For higher frequency recordings of variables with faster dynamics, a higher re-sampling rate might be warranted.

To better condition our inputs, we scale each variable to the $[0, 1]$ interval, using expert-defined ranges. Additionally, we correct for differences in heart rate, respiratory rate, (Fleming et al., 2011) and blood pressure (NHBPEP Working Group 2004) due to age and gender using tables of normal values from large population studies.

2.2 Diagnostic labels

Our work addresses the task of *phenotyping* (Oellrich et al., 2015), that is, recognizing diseases and other relevant clinical conditions, posed as a sequence classification task. However, our methods can be applied to any predictive modeling problem involving sequence data and missing values, such as early prediction of sepsis (Henry et al., 2015) or real-time risk modeling (Wiens et al., 2012). We formulate phenotyping as multilabel classification of diagnostic codes assigned by clinical staff. Our data contain 429 distinct codes (labels) from on an in-house taxonomy used for research and billing, they bear resemblance to the *Ninth Revision of the International Classification of Diseases* (ICD-9) codes (World Health Organization, 2004) commonly used in medical informatics research.

These labels include a wide range of acute conditions, such as acute respiratory distress, congestive heart failure, and sepsis. A full list is given in Appendix A. We focus on the 128 most frequent with at least 50 positive examples. Naturally, the diagnoses are not mutually exclusive. In our data set, the average patient has 2.24 diagnosis codes. Additionally, the base rate frequencies of the diagnoses vary widely (see Appendix A).

3. Recurrent Neural Networks for Multilabel Classification

While our focus in this paper is on missing data, for completeness, we review the LSTM RNN architecture for performing multilabel classification of diagnoses introduced by Lipton et al. (2016). Formally, the problem setup is: Given a series of observations $\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(T)}$, we desire a classifier to generate hypotheses $\hat{\boldsymbol{y}}$ of the true labels $\boldsymbol{y}$, where each input $\boldsymbol{x}^t \in \mathbb{R}^D$ and the output $\hat{\boldsymbol{y}} \in [0, 1]^K$. Here, D denotes the input dimension, K denotes the number of labels, t indexes sequence steps, and for any example, T denotes the length of that sequence.

3.1 LSTM Recurrent Neural Networks

Our proposed RNN uses LSTM memory cells (Hochreiter and Schmidhuber, 1997) with forget gates (Gers et al., 2000) but without peephole connections (Gers et al., 2003). As output, we use a fully connected layer followed by an element-wise logistic activation function σ. We apply *log loss* (binary cross-entropy) as the loss function at each output node.

The following equations give the update for a layer of memory cells $\boldsymbol{h}_l^{(t)}$ where $\boldsymbol{h}_{l-1}^{(t)}$ stands for the previous layer at the same sequence step (a previous LSTM layer or the input $\boldsymbol{x}^{(t)}$) and $\boldsymbol{h}_l^{(t-1)}$ stands for the same layer at the previous sequence step:

$$\boldsymbol{g}_l^{(t)} = \phi(W_l^{\mathrm{gx}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{gh}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{g}})$$

$$\boldsymbol{i}_l^{(t)} = \sigma(W_l^{\mathrm{ix}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{ih}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{i}})$$

$$\boldsymbol{f}_l^{(t)} = \sigma(W_l^{\mathrm{fx}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{fh}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{f}})$$

$$\boldsymbol{o}_l^{(t)} = \sigma(W_l^{\mathrm{ox}}\boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{oh}}\boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{o}})$$

$$\boldsymbol{s}_l^{(t)} = \boldsymbol{g}_l^{(t)} \odot \boldsymbol{i}_l^{(i)} + \boldsymbol{s}_l^{(t-1)} \odot \boldsymbol{f}_l^{(t)}$$

$$\boldsymbol{h}_l^{(t)} = \phi(\boldsymbol{s}_l^{(t)}) \odot \boldsymbol{o}_l^{(t)}.$$

In these equations, σ stands for an element-wise application of the *logistic* function, ϕ stands for an element-wise application of the *tanh* function, and $\odot$ is the Hadamard (element-wise) product. The input, output, and forget gates are denoted by $\boldsymbol{i}$, $\boldsymbol{o}$, and $\boldsymbol{f}$ respectively, while $\boldsymbol{g}$ is the input node and has a *tanh* activation.

3.2 RNN Architecture for Multilabel Classification

The loss at a single sequence step is the average *log loss* calculated across all labels:

$$\mathrm{loss}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{K}\sum_{l=1}^{l=K} -(y_l \cdot \log(\hat{y}_l) + (1 - y_l) \cdot \log(1 - \hat{y}_l)).$$

To overcome the difficulty of learning to pass information across long sequences, we use the *target replication* strategy proposed by Lipton et al. (2016), in which we replicate the static targets at each sequence step (Figure 1), providing a local error signal. This technique is also well-motivated by our problem: we desire to predict accurately even if the sequence were truncated (as in early-warning systems). To calculate loss, we take a convex combination of the final step loss and the average of the losses over predictions $\hat{\boldsymbol{y}}^{(t)}$ at all steps t:

$$\alpha \cdot \frac{1}{T}\sum_{t=1}^{T} \mathrm{loss}(\hat{\boldsymbol{y}}^{(t)}, \boldsymbol{y}^{(t)}) + (1 - \alpha) \cdot \mathrm{loss}(\hat{\boldsymbol{y}}^{(T)}, \boldsymbol{y}^{(T)})$$

where T is the total number of sequence steps and $\alpha \in [0, 1]$ is a hyper-parameter, determining the relative importance of performance on the intermediary vs. final targets. At inference time, we consider only the output at the final step.
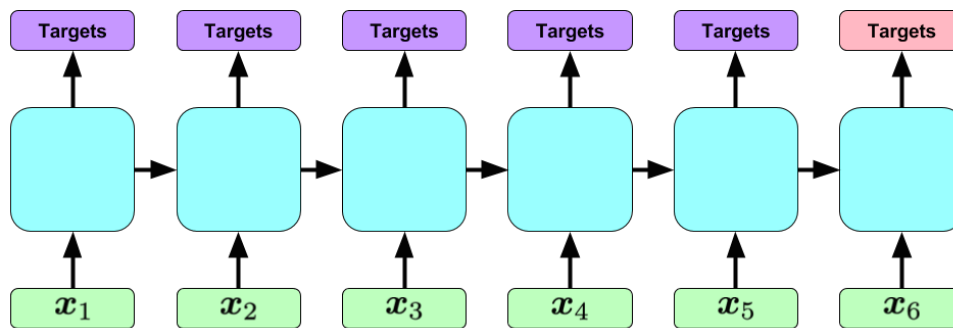
Figure 1: An RNN classification model with *target replication*. The primary target is depicted in red, while intermediate targets are depicted in purple.

4. Missing Data

In this section, we explain our procedures for imputation, missing data indicator sequences, engineering features of missing data patterns.

4.1 Imputation

To address the artifacts of re-sampling we consider two different imputation strategies (forward-filling and zero imputation), as well as direct modeling via indicator variables. Because these approaches are not mutually exclusive, we also evaluate them in combination. Let $x_i^{(t)}$ be the value that results from re-sampling the time series for variable i at time t, and suppose that there are no measurements between hours t and $t-1$ so that $x_i^{(t)}$ is "missing." In our *zero-imputation* strategy, we simply set $x_i^{(t)} := 0$ whenever it is missing. In our *forward-filling* strategy, we impute $x_i^{(t)}$ as follows:

- If there is at least one previously recorded measurement of variable i at a time $t' < t$, we perform forward-filling by setting $x_i^{(t)} := x_i^{(t')}$.

- If there is no previous recorded measurement (or if the variable is missing entirely), then we impute the median estimated over all measurements in the training data.

This strategy is motivated by the intuition that clinical staff record measurements at intervals proportional to rate at which they are believed or observed to change. Heart rate, which can change rapidly, is monitored much more frequently than blood pH. Thus it seems reasonable to assume that a value has changed little since the last time it was measured.

4.2 Learning with Missing Data Indicators

Our *indicator variable* approach to missing data consists of augmenting our inputs with binary variables $m_i^{(t)}$ for every $x_i^{(t)}$, where $m_i^{(t)} := 1$ if $x_i^{(t)}$ is imputed and 0 otherwise. Through their hidden state computations, RNNs can use indicators of missing data to learn arbitrary functions of the past observations and missingness patterns. However, given the same data, linear models can only learn hard substitution rules. To see why, consider a linear model that outputs prediction $f(z)$, where $z = \sum_i w_i \cdot x_i$. With indicator variables,
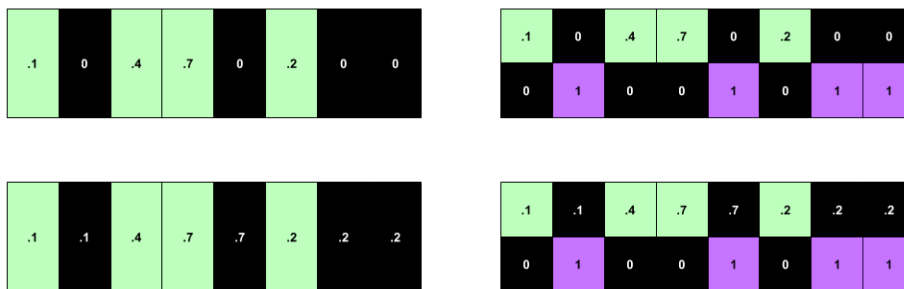
Figure 2: (top left) no imputation or indicators, (bottom left) imputation absent indicators, (top right) indicators but no imputation, (bottom right) indicators and imputation. Time flows from left to right.

we might say that $z = \sum_i w_i \cdot x_i + \sum_i \theta_i \cdot m_i$ where the m_i are corresponding missingness indicators for each x_i and θ_i are the weights for each m_i. If x_i is set to 0 and m_i to 1, whenever the feature x_i is missing, then the impact on the output $\theta_i \cdot m_i = \theta_i$ is exactly equal to the contribution $w_i \cdot x_i^*$ for some $x_i^* = \theta_i/w_i$. In other words, the linear model can only use the indicator in a way that depends neither on the previously observed values $(x_i^1...x_i^{t-1})$, nor any other evidence in the inputs.
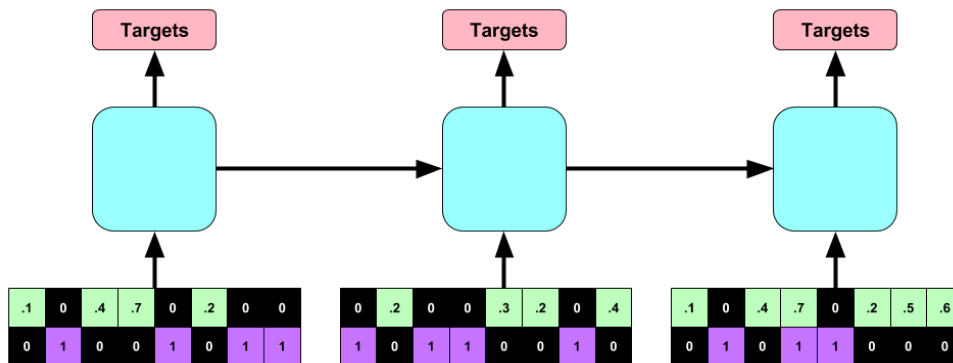
Figure 3: Depiction of RNN zero-filled inputs and missing data indicators.

Note that for a linear model, the impact of a missing data indicator on predictions must be monotonic. In contrast, the RNN might infer that for one patient heart rate is missing because they went for a walk, while for another it might signify an emergency. Also note that even without indicators, the RNN might learn to some extent to recognize *filled-in* vs *real* values. For example, with forward-filling, the RNN could learn to recognize exact repeats. For zero-filling, the RNN could recognize that values set to exactly 0 were likely missing measurements.

4.3 Hand-engineered missing data features

To overcome the limits of the linear model, we also designed features from the indicator sequences. As much as possible, we limited ourselves to features that are simple to calculate,

intuitive, and task-agnostic. The first is a binary indicator for *whether a variable was measured at all*. Additionally, we compute the mean and standard deviation of the missing indicators. The mean captures the frequency with which each variable is measured which carries information about the severity of a patient's condition. The standard deviation, on the other hand, computes a non-monotonic function of frequency that is maximized when a variable is missing exactly 50% of the time. We also compute the frequency with which a variable switches from measured to missing or vice versa across adjacent sequence steps. Finally, we add features that capture *the relative times of the first and last recordings of a variable* , represented as real numbers between 0 and 1 (the number of hours until the measurement, scaled by the length of the interval).

5. Experiments

Our LSTM RNN has 2 hidden layers of 128 LSTM cells each, non-recurrent dropout of 0.5, and ℓ_2^2 weight decay of 10^{-6}. We train on 80% of data, setting aside 10% each for validation and testing. We train each RNN for 100 epochs and then retain the parameters corresponding to the epoch with the lowest validation loss.

We compare the performance of RNNs with stronger versions of the logistic regression and multilayer perceptron (MLP) baselines reported in Lipton et al. (2016). The linear baseline uses an ℓ_2 penalty chosen based on validation performance. The MLP has 3 hidden layers with 500 nodes each and uses rectified linear (ReLU) activations and dropout (with probability of 0.5). This architecture was chosen based on validation performance. We trained the MLP using stochastic gradient descent with momentum.

Because these baselines cannot be applied to variable-length time series, we trained them on one of two different types of fixed size inputs: a concatenation of 12-hour subsequences (of raw measurements, missing indicators, or both) or 12 expressive hand-engineered features designed to capture central tendencies, variability, extremes, and trends: the first and last measurements and their difference, maximum and minimum values, mean and standard deviation, median and 25th and 75th percentiles, and the slope and intercept of least squares line fit. We also computed the 8 missing data features described in section 4.

We improve upon the baselines in Lipton et al. (2016) by computing features over different windows of time, giving them access to greater temporal information and enabling them to better model patterns of missingness. For the raw features, we concatenate three 12-hour subsequences, one each from the beginning, middle, and end of the time series. For shorter time series, these intervals may overlap. This creates a total of $3 \times 12 \times 13 = 468$ raw measurement and indicator features each. We extract hand-engineered features from the entire time series, as well as from three possibly overlapping intervals: the first and last 12 hours and the interval between (for shorter sequences, we instead use the middle 12 hours). This yields a total of $4 \times 12 \times 13 = 624$ and $4 \times 8 \times 13 = 416$ hand-engineered measurement and missing data features, respectively.

We train each model on five different raw inputs: (1) zero-filled missing values, (2) imputed missing values, (2) zero-filled missing values + indicators, (4) imputed missing values + indicators, and (5) indicators-only. In addition we train logistic regression and MLP baselines on three combinations of hand-engineered features Each model is trained and evaluated on the same splits. We evaluate all models with area under the ROC curve

(AUC), F1 score (with threshold chosen on validation data), and precision at 10. For AUC and F1, we report both micro-averaged (calculated across all predictions) and macro-averaged (calculated separately across on each labels) measures to mitigate the weaknesses in each (Lipton et al., 2014). Finally we report precision at 10, whose maximum is 0.2238 because we have on average 2.238 diagnostic codes per patient.

Classification performance for 128 ICU phenotypes

Model	Micro AUC	Macro AUC	Micro F1	Macro F1	P@10
Base Rate	0.7128	0.5	0.1346	0.0343	0.0788
Best Possible	1.0	1.0	1.0	1.0	0.2281
Logistic Regression					
Log Reg - Zeros	0.8108	0.7244	0.2149	0.0999	0.1014
Log Reg - Impute	0.8201	0.7455	0.2404	0.1189	0.1038
Log Reg - Zeros & Indicators	0.8143	0.7269	0.2239	0.1082	0.1017
Log Reg - Impute & Indicators	0.8242	0.7442	0.2467	0.1234	0.1045
Log Reg - Indicators Only	0.7929	0.6924	0.1952	0.0889	0.0939
Multilayer Perceptron					
MLP - Zeros	0.8263	0.7502	0.2344	0.1072	0.1048
MLP - Impute	0.8376	0.7708	0.2557	0.1245	0.1031
MLP - Zeros & Indicators	0.8381	0.7705	0.2530	0.1224	0.1067
MLP - Impute & Indicators	0.8419	0.7805	0.2637	0.1296	0.1082
MLP - Indicators Only	0.8112	0.7321	0.1962	0.0949	0.0947
LSTMs					
LSTM - Zeros	0.8662	0.8133	0.2909	0.1557	0.1176
LSTM - Impute	0.8600	0.8062	0.2967	0.1569	0.1159
LSTM - Zeros & Indicators	**0.8730**	**0.8250**	**0.3041**	**0.1656**	**0.1215**
LSTM - Impute & Indicators	0.8689	0.8206	0.3027	0.1609	0.1196
LSTM - Indicators Only	0.8409	0.7834	0.2403	0.1291	0.1074
Models using Hand-Engineered Features					
Log Reg HE	0.8396	0.7714	0.2708	0.1327	0.1118
Log Reg HE Indicators	0.8472	0.7752	0.2841	0.1376	0.1165
Log Reg HE Indicators Only	0.8187	0.7322	0.2287	0.1001	0.1020
MLP HE	0.8599	0.8052	0.2953	0.1556	0.1168
MLP HE Indicators	0.8669	0.8160	0.2954	0.1610	0.1202
MLP HE Indicators Only	0.8371	0.7682	0.2351	0.1179	0.1028

Table 1: Performance on aggregate metrics for logistic regression (Log Reg), MLP, and LSTM classifiers with and without imputation and missing data indicators.

5.1 Results

We briefly highlight the most noteworthy results from our experiments and defer further discussion until section 7. The best overall model by all metrics (micro AUC of 0.8730) is an LSTM with zero-imputation and missing data indicators, outperforming even the strongest MLP baseline and the LSTM model from (Lipton et al., 2016). While all models benefit from access to missing data indicators, this information confers less benefit to the raw input linear baselines, consistent with theory discussed in subsection 4.2. The performance

of the hand-engineered baselines demonstrates the efficacy of our simple hand-engineered features and the benefit to predictive performance that results from incorporating explicit information about patterns of missingness, whatever the choice of model. One of our most interesting findings is that LSTMs (with or without indicators) appear to perform better with zero-filling than with with imputed values. Even without indicators, the zero-filled LSTM outperforms the LSTM with imputed values. This is remarkable for two reasons. First, this is not true of either baseline; even the MLP performs better with fully imputed data. Second, it suggests that the LSTM may be learning to recognize missing values implicitly as the zeros stand out more than imputed values. For the LSTM, imputation may interfere with its ability to learn an implicitly model missingness patterns. This ability of flexible models to infer missingness may have broader implications for working with data with missing values.

6. Related Work

This work builds upon research relating to missing values and machine learning for medical informatics. The basic RNN methodology for phenotyping derives from Lipton et al. (2016), addressing a dataset and problem described by Che et al. (2015). The methods rely upon LSTM RNNs (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) trained by backpropagation through time (Hinton et al., 2006; Werbos, 1988). A comprehensive perspective on the history and modern applications of recurrent neural networks is provided in (Lipton et al., 2015).

While a long and rich literature addresses pattern recognition with missing data (Cohen and Cohen, 1975; Allison, 2001), most of this literature addresses fixed-length feature vectors, as indicated by reviews of the literature from García-Laencina et al. (2010) and Pigott (2001). Further, while many papers address neural networks for missing values, most of this work focuses on sophisticated imputation strategies, rather than mining missingness as a feature for discriminative learning . Indicator variables for missing data were first proposed long ago by Cohen and Cohen (1975), but we were unable to find papers that have combined missing indicators with recurrent neural networks. Further, only a handful of papers address missing data in the context of RNNS. Bengio and Gingras (1996) demonstrate a scheme by which the RNN learns to fill in the missing values such that the filled-in values minimize output error. In 2001, Parveen and Green (2001) build upon this method to improve automatic speech recognition. Barker et al. (2001) suggests using a mask of indicators, in a scheme for weighting the contribution of reliable vs corrupted data the final prediction. Tresp and Briegel (1998) address missing values by combining a recurrent neural net with a linear state space model to handle uncertainty. This paper may be one of the first to engineer explicit features of missingness patterns in order to improve discriminative performance. Also, to our knowledge, we are the first to harness patterns of missing data to improve the classification of critical care phenotypes.

7. Discussion

Data processing and discriminative learning have often been regarded as separate disciplines. Through this separation of concerns, the complementarity of missing data indicators and training RNNs for classification has been overlooked. This paper proposes that patterns of missing values are an underutilized source of predictive power and that RNNs, unlike

linear models, can effectively mine this signal from sequences of indicator values. Our hypotheses are confirmed by empirical evidence Additionally we introduce and confirm the utility of a simple set of features engineered from the sequence of missingness indicators that can improve performance of linear models and MLPs. These techniques are simple to implement and broadly applicable and seem likely to confer similar benefits for RNNs on other sequential prediction tasks, when data is missing not at random. One example might include financial data, where failures to report accounting details could suggest internal problems at a company.

7.1 The Perils and Inevitability of Modeling Treatment Patterns

For medical applications, the predictive power of missing data raises important philosophical concerns. We train models in the hope that they will make accurate predictions not only on a random test split but also in the real world, once deployed. This hope cuts against the fundamental nature of supervised learning. Unlike reinforcement learners, which learn by interaction with their environment, supervised models, trained offline, cannot account for changes that their deployment might confer upon the real world, possibly invalidating their predictions. Caruana et al. (2015) points this out in the case of a pneumonia risk model that gives lower risk of death to patients who also have asthma. The better outcomes of the asthma patients, as it turns out, owed to the more aggressive treatment they received. The model, if deployed, might be used to choose less aggressive treatment for the patients with both pneumonia and asthma, clearly a sub-optimal course of action.

On the other hand, learning from treatment signal may to some degree be inevitable. Any imputation might leak some information about which values are likely imputed and which aren't. Thus any sufficiently powerful supervised model might catch on to some amount of missingness signal, as was the case in our experiments with the LSTM using zero-filled missing values. Even the physiologic measurements contain information owing to the patterns of treatments. Physiologic signals reflect the medications patients receive and the procedures they undergo. Moreover, often the signal from the patterns of treatments may be intrinsically useful. Doctors already rely on this signal habitually: they read through charts, noting which other doctors have seen a patient, what their opinions might have been, which tests they ordered (or didn't), etc.. While in some circumstances it seems problematic for learning models to rely heavily on patterns of treatment, removing this signal seems not only difficult but also potentially harmful in some cases.

7.2 Complex Models or Complex Features?

Our work also shows that using only simple, re-sampled sequences of data and indicators for missing data, RNNs can achieve state of the art performance classifying clinical time series. The RNNs far outperforms linear models, and MLPs trained on fixed windows. Still, in our experience, even though RNNs and MLPs are both deep learning methods and thus opaque in precisely the same way, there is a strong bias among practitioners toward more familiar models, including MLPs, even when they require substantial feature engineering.

In our experiments, we undertook extensive efforts to engineer features to boost the performance of both linear models and MLPs. Ultimately, while RNNs performed best on raw data, we could approach its performance with an MLP and significantly improve the linear

model by using hand-engineered features and windowing. A question then emerges: how should we evaluate the trade-off between more complex models and more complex features? To the extent that linear models are believed to be more interpretable than neural networks, that notions of interpretability often hinge on the intelligibility of the features. When performance of the linear model (and also the MLP) comes at the price of this intelligibility, we might ask if this tradeoff undermines the linear this chief advantage. Additionally, such a model, while still inferior to the RNN, relies on application-specific features less likely to be useful on other datasets and tasks. In contrast, the RNN, which relies upon the simplest possible futures, with no hand-engineering seems better equipped to generalize to different feature sets, and different tasks. And while the model may seem complex, the inputs remain completely intelligible, opening the possibility to post-hoc interpretations.

References

Paul D Allison. *Missing data*, volume 136. Sage publications, 2001.

Jon Barker, Phil Green, and Martin Cooke. Linking auditory scene analysis and robust asr by missing data techniques. 2001.

Yoshua Bengio and Francois Gingras. Recurrent neural networks for missing or asynchronous data. 1996.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

Zhengping Che, David C. Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 507–516. ACM, 2015.

Jacob Cohen and Patricia Cohen. Applied multiple regression/correlation analysis for the behavioral sciences. 1975.

Susannah Fleming, Matthew Thompson, Richard Stevens, Carl Heneghan, Annette Plüddemann, Ian Maconochie, Lionel Tarassenko, and David Mant. Normal ranges of heart rate and respiratory rate in children from birth to 18 years: A systematic review of observational studies. *The Lancet*, pages 1011–1018, 2011.

Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2): 263–282, 2010.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. Learning precise timing with LSTM recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003.

Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. A targeted real-time early warning score (trewscore) for septic shock. *Science Translational Medicine*, 7 (299 299ra122):1–9, 2015.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Thomas A. Lasko, Joshua C. Denny, and Mia A. Levy. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PLoS ONE*, 8(6):e66341, 06 2013.

Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases*, pages 225–239. Springer, 2014.

Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. Learning to diagnose with lstm recurrent neural networks. *International Conference on Learning Representations*, 2016.

Ben M. Marlin, David C. Kale, Robinder G. Khemani, and Randall C. Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium (IHI)*, 2012.

National High Blood Pressure Education Program Working Group on Children and Adolescents. The fourth report on the diagnosis, evaluation, and treatment of high blood pressure in children and adolescents. *Pediatrics*, 114:555–576, 2004.

Anika Oellrich, Nigel Collier, Tudor Groza, Dietrich Rebholz-Schuhmann, Nigam Shah, Olivier Bodenreider, Mary Regina Boland, Ivo Georgiev, Hongfang Liu, Kevin Livingston, Augustin Luna, Ann-Marie Mallon, Prashanti Manda, Peter N. Robinson, Gabriella Rustici, Michelle Simon, Liqin Wang, Rainer Winnenburg, and Michel Dumontier. The digital revolution in phenotyping. *Briefings in Bioinformatics*, 2015.

Shahla Parveen and P Green. Speech recognition with missing data using recurrent neural nets. In *Advances in Neural Information Processing Systems*, pages 1189–1195, 2001.

Therese D Pigott. A review of methods for missing data. *Educational research and evaluation*, 7(4):353–383, 2001.

Volker Tresp and Thomas Briegel. A solution for missing data in recurrent neural networks with an application to blood glucose prediction. In *Advances in Neural Information Processing Systems (NIPS) 10*, pages 971–977. 1998.

Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988.

Jenna Wiens, Eric Horvitz, and John V Guttag. Patient risk stratification for hospital-associated c. diff as a time-series classification task. In *Advances in Neural Information Processing Systems*, pages 467–475, 2012.

World Health Organization. *International statistical classification of diseases and related health problems*, volume 1. World Health Organization, 2004.

A. Appendix A.

In this appendix we provide per-diagnosis AUC and F1 scores for three representative LSTM models trained with imputed measurements, with imputation plus missing indicators, and with indicators only. By comparing performance on individual diagnoses, we can gain some insight into the relationship between missing values and different conditions. Rows are sorted in descending order based on the F1 score of the imputation plus indicators model. It is worth noting that F1 scores are sensitive to threshold, which we chose in order to optimize per-disease validation F1, sometimes based on a very small number of positive cases. Thus, there are cases where one model will have superior AUC but worse F1.

Classifier Performance on Each Diagnostic Code, Sorted by F1

Condition	Base rate	Imputed Msmt. AUC	Imputed Msmt. F1	Msmt. + indicators AUC	Msmt. + indicators **F1**	Indicators only AUC	Indicators only F1
Diabetes mellitus with ketoacidosis	0.0125	1.0000	0.8889	0.9999	**0.9333**	0.9906	0.7059
Asthma with status asthmaticus	0.0202	0.9384	**0.6800**	0.8907	0.6383	0.8652	0.5417
Scoliosis (idiopathic)	0.1419	0.9143	**0.6566**	0.8970	0.6174	0.8435	0.5235
Tumor, cerebral	0.0917	0.8827	**0.5636**	0.8799	0.5560	0.8312	0.4627
Renal transplant, status post	0.0122	0.9667	0.2963	0.9544	0.4762	0.9490	**0.5600**
Liver transplant, status post	0.0106	0.7534	0.3158	0.8283	**0.4762**	0.8271	0.2581
Acute Respiratory Distress Syndrome	0.0193	0.9696	0.3590	0.9705	**0.4557**	0.9361	0.3333
Developmental delay	0.0876	0.8108	**0.4382**	0.8382	0.4331	0.6912	0.2366
Diabetes insipidus	0.0127	0.9220	0.2727	0.9486	**0.4286**	0.9266	0.4000
End stage renal disease (on dialysis)	0.0241	0.8548	0.2778	0.8800	0.4186	0.9043	**0.4255**
Seizure disorder	0.0816	0.7610	0.3694	0.7937	**0.4059**	0.6431	0.1957
Acute respiratory failure	0.0981	0.8414	0.4128	0.8391	0.3835	0.8358	**0.4542**
Cystic fibrosis	0.0076	0.8628	0.2353	0.8740	**0.3810**	0.8189	0.0000
Septic shock	0.0316	0.8296	0.3363	0.8911	**0.3750**	0.8506	0.1429
Respiratory distress, other	0.0716	0.8411	**0.3873**	0.8502	0.3719	0.7857	0.2143
Intracranial injury, closed	0.0525	0.8886	0.2817	0.9002	**0.3711**	0.8442	0.3208
Arteriovenous malformation	0.0223	0.8620	0.3590	0.8716	**0.3704**	0.8494	0.2857
Seizures, status epilepticus	0.0348	0.8381	**0.4158**	0.8505	0.3704	0.8440	0.3226
Pneumonia due to adenovirus	0.0123	0.8604	0.1250	0.9065	**0.3077**	0.8792	0.1818
Leukemia (acute, without remission)	0.0287	0.8585	0.2783	0.8845	**0.3059**	0.8551	0.2703
Disseminated intravascular coagulopathy	0.0099	0.9556	**0.5000**	0.9532	0.2857	0.9555	0.2500
Septicemia, other	0.0240	0.8586	0.2400	0.8870	**0.2812**	0.7593	0.0000
Bronchiolitis	0.0162	0.9513	0.2667	0.9395	**0.2703**	0.8826	0.1778
Congestive heart failure	0.0133	0.8748	0.1429	0.8756	**0.2703**	0.8326	0.1364
Upper airway obstruction (UAO), other	0.0378	0.8206	**0.2564**	0.8573	0.2542	0.8350	0.1964
Diabetes mellitus type I, stable	0.0064	0.7105	0.0000	0.9625	0.2500	0.9356	**0.3333**
Cerebral palsy (infantile)	0.0262	0.8230	**0.2609**	0.8359	0.2500	0.6773	0.0980
Coagulopathy	0.0131	0.7501	0.1111	0.8098	**0.2449**	0.8548	0.1667
UAO, ENT surgery, post-status	0.0302	0.9059	**0.4058**	0.8733	0.2400	0.8364	0.1975
Hypertension, systemic	0.0169	0.8740	0.2105	0.8831	0.2388	0.8216	**0.2857**
Acute renal failure, unspecified	0.0191	0.9242	0.2381	0.9510	0.2381	0.9507	**0.3291**
Trauma, vehicular	0.0308	0.8673	0.2105	0.8649	**0.2381**	0.8022	0.1395
Hepatic failure (acute necrosis of liver)	0.0176	0.8489	0.2222	0.9239	**0.2308**	0.8598	0.1935
Craniosynostosis (anomalies of skull)	0.0064	0.7824	0.0000	0.9267	**0.2286**	0.8443	0.0315
Prematurity (<37 weeks gestation)	0.0321	0.7520	0.1548	0.7542	**0.2245**	0.7042	0.1266
Hydrocephalus, other (congenital)	0.0381	0.7118	0.2099	0.7500	**0.2241**	0.7065	0.1961
Pneumothorax	0.0134	0.8220	0.1176	0.7957	0.2188	0.7552	**0.3243**
Congenital muscular dystrophy	0.0121	0.8427	**0.2500**	0.8491	0.2143	0.7460	0.0800
Cardiomyopathy (primary)	0.0191	0.7508	0.1290	0.6057	**0.2143**	0.6372	0.1818
Pulmonary edema	0.0076	0.8839	0.0769	0.8385	**0.2105**	0.8071	0.0870

Table 2: AUC and F1 scores for individual diagnostic codes.

Classifier Performance on Each Diagnostic Code, Sorted by F1

Condition	Base rate	Imputed Msmt.		Msmt. + indicators		Indicators only	
		AUC	F1	AUC	**F1**	AUC	F1
(Acute) pancreatitis	0.0106	0.8712	0.0769	0.9512	**0.2000**	0.8182	0.0571
Tumor, disseminated or metastatic	0.0180	0.7178	0.0938	0.7415	**0.1967**	0.6837	0.1062
Hematoma, intracranial (ICH / IVH)	0.0299	0.7724	**0.2278**	0.8249	0.1892	0.7518	0.1474
Neutropenia (agranulocytosis)	0.0108	0.8285	0.0000	0.8114	0.1852	0.8335	**0.2609**
Arrhythmia, other	0.0087	0.8536	0.0000	0.8977	**0.1818**	0.8654	0.0000
Child abuse, suspected	0.0065	0.9544	**0.2222**	0.8642	0.1818	0.8227	0.0870
Encephalopathy, hypoxic or ischemic	0.0116	0.8242	0.1429	0.8571	**0.1818**	0.8009	0.0800
Epidural hematoma	0.0098	0.7389	0.0455	0.8233	**0.1818**	0.7936	0.1000
Tumor, gastrointestinal	0.0100	0.8112	0.1429	0.8636	**0.1778**	0.8732	0.0984
Craniofacial malformation	0.0133	0.8707	**0.2667**	0.8514	0.1778	0.6928	0.2286
Gastroesophageal reflux	0.0182	0.7571	**0.1818**	0.8554	0.1690	0.7739	0.1600
Pneumonia, bacterial (pneumococcal)	0.0186	0.8876	0.1333	0.8806	**0.1600**	0.8616	0.0000
Pneumonia, undetermined	0.0179	0.8323	0.1481	0.8269	**0.1583**	0.7772	0.0947
Cerebral edema	0.0059	0.8275	0.0000	0.9469	**0.1538**	0.9195	0.1500
Pneumonia due to inhalation	0.0078	0.7917	0.1111	0.8602	**0.1538**	0.8268	0.0566
Metabolic or endocrine disorder, other	0.0095	0.7718	0.0364	0.6929	0.1538	0.6319	**0.2000**
Disorder of kidney and ureter, other	0.0204	0.8486	**0.2857**	0.8650	0.1500	0.8238	0.2500
Urinary tract infection	0.0137	0.7478	0.1154	0.7402	**0.1481**	0.7229	0.0588
Subdural hematoma	0.0147	0.8270	**0.1449**	0.8884	0.1429	0.8190	0.0476
Near drowning	0.0068	0.8296	0.0741	0.7917	**0.1404**	0.6897	0.0397
Cardiac arrest, outside hospital	0.0118	0.8932	0.0976	0.8791	**0.1379**	0.8881	0.0556
Pleural effusion	0.0113	0.8549	0.1081	0.8186	**0.1351**	0.7605	0.1151
Bronchopulmonary dysplasia	0.0252	0.8309	**0.1915**	0.7952	0.1304	0.8503	0.1203
Hyponatremia	0.0056	0.5707	0.0187	0.7398	**0.1176**	0.8775	0.0000
Suspected septicemia, rule out	0.0143	0.7378	0.0923	0.7402	**0.1029**	0.6769	0.0000
Thrombocytopenia	0.0112	0.7381	0.0822	0.7857	**0.1026**	0.8585	0.0800
(Benign) intracranial hypertension	0.0099	0.8494	0.0000	0.9018	0.1020	0.8586	**0.1224**
Pericardial effusion	0.0055	0.8997	0.0870	0.9085	**0.1017**	0.9000	0.0714
Pulmonary contusion	0.0068	0.9029	0.0606	0.8831	**0.0984**	0.8197	0.0225
Surgery, gastrointestinal	0.0104	0.6705	0.0714	0.6666	**0.0976**	0.5545	0.0233
Respiratory Arrest	0.0062	0.8404	0.0000	0.8741	**0.0952**	0.8127	0.0444
Trauma, abdominal	0.0105	0.7426	**0.1667**	0.8623	0.0930	0.6991	0.0426
Atrial septal defect	0.0107	0.7766	0.0727	0.7765	**0.0909**	0.7197	0.0000
Genetic abnormality	0.0557	0.6629	**0.1324**	0.6470	0.0876	0.5705	0.1165
Arrhythmia, ventricular	0.0062	0.8532	0.0303	0.8703	0.0870	0.8182	**0.1250**
Hematologic disorder, other	0.0114	0.6736	0.0800	0.6898	**0.0870**	0.8074	0.0800
Asthma, stable	0.0171	0.7010	**0.0925**	0.6607	0.0870	0.5907	0.0741
Neurofibromatosis	0.0079	0.8022	0.0469	0.7984	**0.0816**	0.7388	0.0160
Tumor, bone	0.0090	0.8830	0.0727	0.8174	**0.0800**	0.7649	0.0417
Shock, hypovolemic	0.0088	0.7703	0.0000	0.8433	**0.0741**	0.8040	0.0000
Gastrointestinal bleed, other	0.0064	0.8325	0.0541	0.7974	0.0741	0.7996	**0.0909**

Classifier Performance on Each Diagnostic Code, Sorted by F1

Condition	Base rate	Imputed Msmt. AUC	Imputed Msmt. F1	Msmt. + indicators AUC	Msmt. + indicators F1	Indicators only AUC	Indicators only F1
Chromosomal abnormality	0.0173	0.8047	0.1034	0.7197	0.0714	0.6300	**0.1600**
Encephalopathy, other	0.0093	0.8265	0.1250	0.8736	0.0688	0.8335	**0.1250**
Respiratory syncytial virus	0.0130	0.8876	**0.2857**	0.9145	0.0645	0.8716	0.0930
(Hereditary) hemolytic anemia, other	0.0088	0.7582	0.0548	0.8544	**0.0645**	0.9125	0.0513
Obstructive sleep apnea	0.0185	0.7564	0.0613	0.8200	0.0645	0.8087	**0.1111**
Apnea, central	0.0142	0.7871	**0.1600**	0.8134	0.0625	0.8051	0.0000
Neuromuscular, other	0.0132	0.7163	0.0452	0.7069	**0.0619**	0.6484	0.0392
Anemia, acquired	0.0056	0.7378	**0.1017**	0.7596	0.0615	0.8129	0.0519
Meningitis, bacterial	0.0070	0.4431	0.0000	0.7676	**0.0606**	0.5480	0.0000
Trauma, long bone injury	0.0096	0.8757	0.0952	0.9085	0.0597	0.7946	**0.1176**
Bowel (intestinal) obstruction	0.0104	0.7512	**0.0984**	0.6559	0.0597	0.6936	0.0424
Neurologic disorder, other	0.0288	0.7628	**0.1481**	0.6978	0.0588	0.5971	0.0769
Panhypopituitarism	0.0057	0.7763	0.0000	0.7724	**0.0571**	0.6415	0.0000
Thyroid dysfunction	0.0072	0.6310	0.0369	0.6420	**0.0541**	0.6661	0.0000
Coma	0.0056	0.6483	**0.1250**	0.6823	0.0513	0.7155	0.0000
Spinal cord lesion	0.0133	0.7298	**0.0585**	0.7052	0.0488	0.8168	0.0414
Pneumonia, other (mycoplasma)	0.0188	0.8589	**0.1613**	0.8792	0.0476	0.8424	0.1164
Trauma, blunt	0.0065	0.9156	**0.0513**	0.8138	0.0469	0.7426	0.0177
Surgery, thoracic	0.0058	0.7405	0.0000	0.6948	0.0469	0.6087	**0.0909**
Neuroblastoma	0.0059	0.6526	0.0306	0.7268	**0.0360**	0.7775	0.0346
Obesity	0.0098	0.7503	0.0365	0.6814	0.0351	0.6647	**0.0667**
Obstructed ventriculoperitoneal shunt	0.0073	0.6824	0.0267	0.7114	0.0331	0.7516	**0.0667**
Ventricular septal defect	0.0119	0.6641	**0.1081**	0.5680	0.0294	0.5593	0.0444
Croup Syndrome, UAO	0.0069	0.9418	0.2222	0.9834	0.0000	0.9682	**0.2222**
Sickle-cell anemia, unspecified	0.0080	0.6262	0.0000	0.9627	0.0000	0.8661	**0.1250**
Biliary atresia	0.0063	0.9383	**0.2667**	0.9164	0.0000	0.7589	0.0714
Metabolic acidosis (¡7.1)	0.0083	0.9475	**0.1818**	0.9046	0.0000	0.9143	0.1538
Immunologic disorder, other	0.0094	0.9539	**0.1500**	0.8868	0.0000	0.8969	0.1212
Pulmonary hypertension, other	0.0112	0.9259	**0.2500**	0.8826	0.0000	0.8098	0.0000
Trauma, chest	0.0051	0.9261	0.0000	0.8818	0.0000	0.7820	0.0000
Spinal muscular atrophy	0.0052	0.9666	0.0000	0.8658	0.0000	0.8362	0.0000
Trauma, unspecified	0.0065	0.7153	**0.1481**	0.8657	0.0000	0.8224	0.0594
Bone marrow transplant, status post	0.0097	0.8161	**0.5217**	0.8562	0.0000	0.8505	0.1695
Surgery, orthopaedic	0.0180	0.7839	**0.1029**	0.8192	0.0000	0.7331	0.0000
Gastrointestinal bleed, upper	0.0063	0.8388	0.0000	0.8078	0.0000	0.7256	0.0000
Arrhythmia, supraventricular tachy.	0.0055	0.8178	**0.0385**	0.7867	0.0000	0.8199	0.0000
Congenital central alveolar hypovent.	0.0057	0.7067	0.0000	0.7716	0.0000	0.7282	0.0000
Tetralogy of fallot	0.0061	0.5759	0.0000	0.7614	0.0000	0.7637	0.0000
Cardiac disorder, other	0.0071	0.7229	**0.0519**	0.7552	0.0000	0.6287	0.0000
Hydrocephalus, shunt failure	0.0083	0.7715	0.0000	0.7542	0.0000	0.7986	**0.0635**
Cerebral infarction (CVA)	0.0058	0.6766	0.0000	0.7495	0.0000	0.7148	**0.1333**
Congenital heart disorder, other	0.0084	0.7590	0.0000	0.7277	0.0000	0.7803	**0.0583**
Gastrointestinal disorder, other	0.0139	0.6755	0.0336	0.6821	0.0000	0.6465	**0.1026**
Aspiration	0.0072	0.6727	**0.0533**	0.6734	0.0000	0.6792	0.0333
Dehydration	0.0105	0.7356	**0.0690**	0.6636	0.0000	0.5899	0.0000
Tumor, thoracic	0.0077	0.6931	**0.0513**	0.6249	0.0000	0.6815	0.0292
UAO, extubation, status post	0.0085	0.8295	**0.0672**	0.6063	0.0000	0.6128	0.0000

B. Appendix B.

In this appendix, we present information about the sampling rates and missingness characteristics of our 13 variables. The first column lists the average number of measurements per hour in all episodes with at least one measurement (excluding episodes where the variable is missing entirely). The second column lists the fraction of episodes in which the variable is missing completely (there are zero measurements). The third column lists the missing rate in the resulting re-sampled sequences.

Variable	Measurements/hour	Missing entirely	Frac. missing
Diabstolic blood pressure	0.5162	0.0135	0.1571
Systolic blood pressure	0.5158	0.0135	0.1569
Peripheral capillary refall rate	1.0419	0.0140	0.5250
End-tidal CO_2	0.9318	0.5710	0.5727
Fraction inspired O_2	1.3004	0.1545	0.7873
Total glasgow coma scale	1.0394	0.0149	0.5250
Glucose	1.4359	0.1323	0.9265
Heart rate	0.2477	0.0133	0.0329
pH	1.4580	0.3053	0.9384
Respiratory rate	0.2523	0.0147	0.0465
Pulse oximetry	0.1937	0.0022	0.0326
Temperature	1.0210	0.0137	0.5235
Urine output	1.1160	0.0353	0.5980

Table 3: Sampling rates and missingness statistics for all 13 features.