Recurrent Neural Networks for Multivariate Time Series with Missing Values

Zhengping Che
University of Southern California
Los Angeles, CA 90089
zche@usc.edu

Sanjay Purushotham
University of Southern California
Los Angeles, CA 90089
spurusho@usc.edu

Kyunghyun Cho
New York University
New York, NY 10012
kyunghyun.cho@nyu.edu

David Sontag
New York University
New York, NY 10012
dsontag@cs.nyu.edu

Yan Liu
University of Southern California
Los Angeles, CA 90089
yanliu.cs@usc.edu

Abstract

Many multivariate time series data in practical applications, such as health care, geoscience, and biology, are characterized by a variety of missing values. It has been noted that the missing patterns and values are often correlated with the target labels, a.k.a., missingness is *informative*, and there is significant interest to explore methods which model them for time series prediction and other related tasks. In this paper, we develop novel deep learning models based on Gated Recurrent Units (GRU), a state-of-the-art recurrent neural network, to handle missing observations. Our model takes two representations of missing patterns, i.e., *masking* and *time duration*, and effectively incorporates them into a deep model architecture so that it not only captures the long-term temporal dependencies in time series, but also utilizes the missing patterns to improve the prediction results. Experiments of time series classification tasks on real-world clinical datasets (MIMIC-III, PhysioNet) and synthetic datasets demonstrate that our models achieve state-of-art performance on these tasks and provide useful insights for time series with missing values.

1 Introduction

Multivariate time series data are ubiquitous in many practical applications ranging from health care, geoscience, astronomy, to biology and others. They often inevitably carry missing observations due to various reasons, such as medical events, saving costs, anomalies, inconvenience and so on. It has been noted that these missing values are usually *informative missingness* [22], i.e., the missing values and patterns provide rich information about target labels in supervised learning tasks (e.g, time series classification). To illustrate the idea, we show some examples from two real world health care datasets (MIMIC-III, PhysioNet) in Figure 1. We plot the Pearson correlation coefficient between variable missing rates and the labels (mortality and ICD-9 diagnoses). We observe that the missing rate is correlated with the labels, and the variables with low missing rate are usually highly correlated with the labels, demonstrating the usefulness of missingness patterns in solving a prediction task.

In the past decades, various approaches have been developed to address the missing values [23]. A simple solution is to omit the missing data and perform analysis only on the observed data. A variety of solutions have been developed to fill in the missing values, such as smoothing or interpolation [13], spectral analysis [17], kernel methods [20], multiple imputation [28], and the EM algorithm [7]. [23] and references therein provide a good review on missing data solutions. However, existing solutions often result in a two-step process where imputations are disparate from the prediction models and

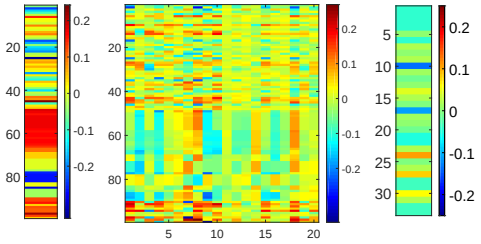


Figure 1: Correlations between missing rate and labels in real world health care datasets. Left: MIMIC-III mortality label; middle: MIMIC-III ICD-9 diagnoses labels; right: Physionet 2012 mortality label. x-axis, target label(s); y-axis, input variables; color: correlation value. Please refer to supplementary materials for more details about this figure.

missing patterns are not effectively explored, thus leading to suboptimal performance in prediction and analysis [27].

In the meanwhile, Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) [9] and Gated Recurrent Unit (GRU) [4], have shown to achieve the state-of-the-art results in many applications with time series or sequential data, including machine translation [1, 25] and speech recognition [8]. RNNs enjoy several good properties such as strong prediction performance as well as the ability to capture long-term temporal dependencies and variable-length observations. RNNs for missing data has been studied in earlier works [3, 26, 18] and applied for speech recognition and blood-glucose prediction. However, there has not been works which directly model missing patterns into RNN for time series classification problems. Exploiting the power of RNNs along with the *informativeness* of missing patterns is a new promising venue to effectively model multivariate time series and is the main motivation behind our work.

In this paper, we develop novel deep learning models based on Gated Recurrent Units (GRU) to effectively exploit two types of informative missingness patterns, i.e., *masking* and *time duration*. Masking informs the model which inputs are observed (or missing), while time duration encapsulates the input observation patterns. Our model captures the observations and their dependencies by applying masking and time duration (using a decay term) to the inputs and network states of GRU, and can be jointly trained using backpropagation. Thus, our models not only can capture the long-term temporal dependencies of time series observations but also can utilize the missing patterns to improve the prediction results. Empirical experiments on real-world clinical datasets as well as synthetic datasets demonstrate that our proposed models outperform strong deep learning models built on GRU with imputation as well as other strong baselines. These experiments show that our proposed method is suitable for many time series classification problems with missing data, and in particular is readily applicable to the predictive tasks in emerging healthcare applications. Moreover, our method also provides useful insights into more general research challenges of time series analysis with missing data beyond classification tasks, including: (1) Effective solutions to characterize the missing patterns of missing-at-not-random time series, such as masking and time duration modeling; (2) A general deep learning framework to handle time series with missing data; (3) An interesting analysis on relationship between missingness and the outcomes for the proposed models on a varieties of datasets.

2 RNN models for time series with missing variables

2.1 Notations

We denote a multivariate time series with D variables of length T as $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)^{\mathrm{T}} \in \mathbb{R}^{T \times D}$, where $\boldsymbol{x}_t \in \mathbb{R}^D$ represents the t-th observations/measurements of all variables and x_t^d denotes the measurement of d-th variable of $\boldsymbol{x}_t$. Let $s_t \in \mathbb{R}$ denote the time-stamp when the t-th observation is obtained and we assume that the first observation is made at time $t = 0$ ($s_1 = 0$). A time series $\boldsymbol{X}$ could have missing values. We introduce a *mask vector* $\boldsymbol{m}_t \in \{0, 1\}^D$ to denote which variables are missing at time step t. The mask vector for $\boldsymbol{x}_t$ is given by

$$m_t^d = \begin{cases} 1, & \text{if } x_t^d \text{ is observed} \\ 0, & \text{otherwise} \end{cases}$$

X: Input time series (2 variables);
s: Timestamps for X;
M: Masking for X;
Δ: Time duration for X.

$$X = \begin{bmatrix} 47 & 49 & NA & 40 & NA & 43 & 55 \\ NA & 15 & 14 & NA & NA & NA & 15 \end{bmatrix}$$

$$s = [0 \quad 0.1 \quad 0.6 \quad 1.6 \quad 2.2 \quad 2.5 \quad 3.1]$$

$$M = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Delta = \begin{bmatrix} 0.0 & 0.1 & 0.5 & 1.5 & 0.6 & 0.9 & 0.6 \\ 0.0 & 0.1 & 0.5 & 1.0 & 1.6 & 1.9 & 2.5 \end{bmatrix}$$

Figure 2: An example sequence of measurement vectors $\boldsymbol{x}_t$, time stamps s_t, mask vectors $\boldsymbol{m}_t$ and durations δ_t.

For each variable d, we also maintain the *time duration* since its last observation, and it is denoted by $\delta_t^d \in \mathbb{R}$ and given as:

$$\delta_t^d = \begin{cases} s_t - s_{t-1} + \delta_{t-1}^d, & t > 1, m_{t-1}^d = 0 \\ s_t - s_{t-1}, & t > 1, m_{t-1}^d = 1 \\ 0, & t = 1 \end{cases}$$

An example of these notations is illustrated in Figure 2. We also denote the missing rate for a variable d as $p_{\boldsymbol{X}}^d$ and it is calculated as $p_{\boldsymbol{X}}^d = 1 - \frac{1}{T}\sum_{t=1}^{T} m_t^d$. In this paper, we are interested in the time series classification problem, where we predict the labels l_n given the time series data $\mathcal{D}$, where $\mathcal{D} = \{(\boldsymbol{X}_n, \boldsymbol{s}_n, \boldsymbol{M}_n, \Delta_n, l_n)\}_{n=1}^N$, and $\boldsymbol{X}_n = \left[\boldsymbol{x}_1^{(n)}, \ldots, \boldsymbol{x}_{T_n}^{(n)}\right]$, $\boldsymbol{s}_n = \left[s_1^{(n)}, \ldots, s_{T_n}^{(n)}\right]$, $\boldsymbol{M}_n = \left[\boldsymbol{m}_1^{(n)}, \ldots, \boldsymbol{m}_{T_n}^{(n)}\right]$, $\Delta_n = \left[\boldsymbol{\delta}_1^{(n)}, \ldots, \boldsymbol{\delta}_{T_n}^{(n)}\right]$, and $l_n \in \{1, \ldots, L\}$.

2.2 Recurrent neural networks for time-series classification

In this paper, we investigate the use of recurrent neural networks (RNN) for time-series classification, as their recursive formulation allows it to handle variable-length sequences naturally. Moreover, RNN shares the same parameters across all time steps which greatly reduces the total number of parameters we need to learn. Among different variants of the RNN, we specifically consider an RNN with gated recurrent units [4, 6].

The structure of GRU is shown in Figure 3(a). GRU has a reset gate r_t^j and an update gate z_t^j for each of the hidden state h_t^j to control. At each time t, the update functions are shown as follows:

$$\begin{array}{ll} \boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) & \boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right) \\ \tilde{\boldsymbol{h}}_t = \tanh\left(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}\right) & \boldsymbol{h}_t = (\boldsymbol{1} - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t \end{array}$$

where matrices $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}, \boldsymbol{U}_z, \boldsymbol{U}_r, \boldsymbol{U}$ and vectors $\boldsymbol{b}_z, \boldsymbol{b}_r, \boldsymbol{b}$ are model parameters. σ is an element-wise sigmoid function, and we use $\odot$ for element-wise multiplication. This formulation assumes that all the variables are observed.

2.3 RNN baseline approaches

There are two straightforward approaches to using an RNN for time series data with missing variables. The first, and perhaps most naive, approach is to preprocess the time series so that it does not have any missing variables when presented to an RNN. We describe three such approaches in this section.

GRU-0 We may simply replace each missing observation with the mean of the variable across the training examples, i.e.,

$$x_t^d \leftarrow m_t^d x_t^d + (1 - m_t^d)\tilde{x}^d, \tag{1}$$

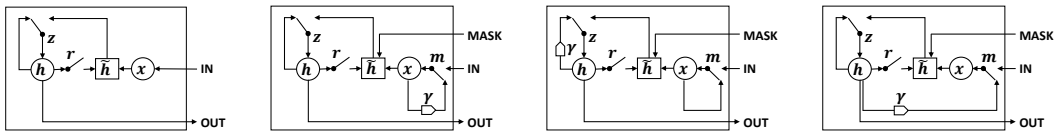
where $\tilde{x}^d = \sum_{n=1}^N \sum_{t=1}^{T_n} m_{t,n}^d x_{t,n}^d \Big/ \sum_{n=1}^N \sum_{t=1}^{T_n} m_{t,n}^d$.

GRU-f We can exploit the temporal structure in each time series. That is, we assume that any missing measurement is same as the last measurement, i.e.,

$$x_t^d \leftarrow m_t^d x_t^d + (1 - m_t^d)x_{t'}^d, \tag{2}$$

where $t' < t$ is the last time the d-th variable was observed. This is the forward imputation model. Our preliminary experiments showed that forward imputation works better than imputating missing values based on interpolation. If the first few measurements are missing, we do backward imputation. If all the measurements of a variable are missing, then we impute with empirical mean.

GRU-xmd Instead of explicitly imputing missing values, we may simply indicate which variables are missing as a part of input to a GRU-RNN. We do this by concatenating the measurement, mask and duration vectors: $\boldsymbol{x}_t^{(n)} \leftarrow \left[\boldsymbol{x}_t^{(n)}; \boldsymbol{m}_t^{(n)}; \boldsymbol{\delta}_t^{(n)}\right]$, where $\boldsymbol{x}_t^{(n)}$ is from Equation 1 or 2.



(a) GRU (b) GRU-DM (c) GRU-DS (d) GRU-DI

Figure 3: Graphical illustrations of (a) the original GRU, (b, c) GRU with the trainable decay rates and (d) GRU with the dynamic imputation.

2.4 Trainable decay models

Let us begin by relaxing the assumption made for the forward imputation model. Instead of using the last observation as it is, we may decay it over time toward the empirical mean (which we take as a *default* configuration). There are two things to be considered when decaying variables. First, we want the rate at which each variable decays to differ from the other variables based on the underlying nature of the variable. Second, the decay rate should be indicative of missingness patterns which are informative (as we have shown earlier). In short, we aim at modeling decay rates to be learned rather than fixed a priori, based on the missingness pattern. We model a vector of decay rates $\boldsymbol{\gamma}$ as

$$\boldsymbol{\gamma}_t = \exp\left\{-\max\left(\mathbf{0}, \boldsymbol{W}_\gamma \boldsymbol{\delta_t} + \boldsymbol{b}_\gamma\right)\right\}, \tag{3}$$

where $\boldsymbol{W}_\gamma$ and $\boldsymbol{b}_\gamma$ are model parameters that we train jointly with all the other parameters of the RNN. We chose the exponentiated negative rectifier in order to keep each decay rate in a reasonable rate between 0 and 1. We however note that it is possible to use other formulations such as a sigmoid function instead of the exponentiated negative rectifier, as long as the resulting decay rate $\gamma_t^j \in [0, 1]$.

GRU-DM: Input decay This trainable decay scheme can be readily applied to the measurement vector by

$$x_t^d \leftarrow m_t^d x_t^d + (1 - m_t^d)\gamma_t^d x_{t'}^d + (1 - m_t^d)(1 - \gamma_t^d)\tilde{x}^d \tag{4}$$

where $x_{t'}^d$ is the last observation of the d-th variable ($t' < t$) and $\tilde{x}^d$ is the empirical mean of the d-th variable. When decaying the input variable directly, we constrain $\boldsymbol{W}_\gamma$ to be diagonal, which effectively makes the decay rate of each variable independent from the others'.

GRU-DS: Hidden decay As the decay rates are trained together with the whole GRU-RNN, we can instead decay the hidden state of the RNN. Intuitively, this has an effect of decaying the features rather than raw input variables. This is implemented by decaying the previous hidden state $\boldsymbol{h}_{t-1}$ before computing a new hidden state $\boldsymbol{h}_t$:

$$\boldsymbol{h}_{t-1} \leftarrow \boldsymbol{\gamma}_t \odot \boldsymbol{h}_{t-1}, \tag{5}$$

in which case we do not constrain $\boldsymbol{W}_\gamma$ to be diagonal.

2.5 GRU-DI: Goal-oriented imputation model

We may alternatively let the GRU-RNN predict the missing values in the next timestep on its own. When missing values occur only during test time, we simply train the GRU-RNN to predict the measurement vector of the next time step as a language model [16] and use it to fill the missing values during test time. This is unfortunately not applicable for some time series applications such as in healthcare domain, which also have missing data during training.

Instead, we propose here to view missing values as latent variables in a probabilistic graphical model. Given a timeseries $\boldsymbol{X}$, we denote all the missing variables by $\mathcal{M}_{\boldsymbol{X}}$ and all the observed ones by $\mathcal{O}_{\boldsymbol{X}}$. Then, training a time-series classifier with missing variables becomes equivalent to maximizing the marginalized log-conditional probability of a correct label l, i.e., $\log p(l|\mathcal{O}_{\boldsymbol{X}})$.

The exact marginalized log-conditional probability is however intractable to compute, and we instead maximize its lowerbound:

$$\log p(l|\mathcal{O}_{\boldsymbol{X}}) = \log \sum_{\mathcal{M}_{\boldsymbol{X}}} p\left(l|\mathcal{M}_{\boldsymbol{X}}, \mathcal{O}_{\boldsymbol{X}}\right) p\left(\mathcal{M}_{\boldsymbol{X}}|\mathcal{O}_{\boldsymbol{X}}\right) \geq \mathbb{E}_{\mathcal{M}_{\boldsymbol{X}} \sim p(\mathcal{M}_{\boldsymbol{X}}|\mathcal{O}_{\boldsymbol{X}})} \log p\left(l|\mathcal{M}_{\boldsymbol{X}}, \mathcal{O}_{\boldsymbol{X}}\right),$$

where we assume the distribution over the missing variables at each time step is only conditioned on all the previous observations:

$$p\left(\mathcal{M}_{\boldsymbol{X}}|\mathcal{O}_{\boldsymbol{X}}\right) = \prod_{t=1}^{T} \prod_{1 \leq d \leq D}^{m_t^d=1} p(x_t^d|\boldsymbol{x}_{1:(t-1)}, \boldsymbol{m}_{1:(t-1)}, \boldsymbol{\delta}_{1:(t-1)}). \tag{6}$$

Although this lowerbound is still intractable to compute exactly, we can approximate it by Monte Carlo method, which amounts to sampling the missing variables at each time as the RNN reads the input sequence from the beginning to the end, such that

$$x_t^d \leftarrow m_t^d x_t^d + (1 - m_t^d)\tilde{x}_t^d, \tag{7}$$

where $\tilde{\boldsymbol{x}}_t \sim x_t^d|\boldsymbol{x}_{1:(t-1)}, \boldsymbol{m}_{1:(t-1)}, \boldsymbol{\delta}_{1:(t-1)}$.

By further assuming that $\tilde{\boldsymbol{x}}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2\right)$ where $\boldsymbol{\mu}_t = \boldsymbol{\gamma}_t \odot (\boldsymbol{W}_x \boldsymbol{h}_{t-1} + \boldsymbol{b}_x)$ and $\boldsymbol{\sigma}_t = \mathbf{1}$, we can use a reparametrization technique widely used in stochastic variational inference [12, 21] to estimate the gradient of the lowerbound efficiently. During the test time, we simply use the mean of the missing variable, i.e., $\tilde{\boldsymbol{x}}_t = \boldsymbol{\mu}_t$, as we have not seen any improvement from Monte Carlo approximation in our preliminary experiments. We view this approach as a goal-oriented imputation method, and refer to this approach as GRU-DI. The whole model is trained to minimize the classification cross-entropy error ℓ_{log_loss} and we take the negative log likelihood of the observed values as a regularizer.

$$\ell = \ell_{log\_loss} + \lambda \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T_n} \sum_{t=1}^{T_n} \frac{\sum_{d=1}^{D} m_t^d \cdot \log p(x_t^d|\mu_t^d, \sigma_t^d)}{\sum_{d=1}^{D} m_t^d} \tag{8}$$

3 Experiments

We demonstrate the performance of our proposed models on one synthetic and two real-world healthcare datasets (MIMIC-III, PhysioNet) and compare it to several strong machine learning and deep learning approaches in classification tasks. We study the impact of informative missingness on the model performance. We also evaluate our models for different settings such as early prediction and different dataset sizes.

3.1 Dataset descriptions and experimental design

Table 1: Dataset statistics

	MIMIC-III	**PhysioNet2012**	**Gesture**
# of samples (N)	19714	4000	378
# of variables (D)	99	33	23
Mean of # of time steps	35.89	68.91	21.42
Max. of # of time steps	150	155	31
Mean variable missing rate	0.9621	0.8225	N/A

To evaluate our proposed framework, we ran a series of prediction experiments on three datasets. Statistics of these datasets are shown in Table 1. For each dataset, we only consider time steps when at least one measurement is available.

Gesture phase segmentation data This UCI dataset [15] has multivariate time series features with 5 different gesticulations. It is regularly sampled and has no missing values. We extracted 378 time series and randomly introduced missing values to generate 4 synthetic datasets. The missing rates in these synthetic datasets are the same (around 50%) but have different correlations with the ground-truth labels. We use these datasets to study the impact of modeling missingness patterns in our models.

Physionet challenge 2012 data This dataset, from *PhysioNet Challenge 2012* [24], is a publicly available collection of multivariate clinical time series from 8000 ICU records. Each record is a multivariate time series of roughly 48 hours and contains 33 variables such as *Albumin, heart-rate, glucose* etc. We used *Training Set A* subset in our experiments since outcomes (such as in-hospital mortality labels) are publicly available only for this subset. We conduct the following two prediction tasks on this dataset.

- Mortality (Phy-Mor) task – Predict whether the patient dies in the hospital. There are 554 patients with positive mortality label. We treat this as a binary classification problem.
- All 4 (Phy-4tasks) tasks – Predict 4 tasks: in-hospital mortality (mortality), length-of-stay less than 3 days (los<3), whether the patient had a cardiac condition (cardiac), and whether the patient was recovering from surgery (surgery). We consider this as a multi-task prediction problem.

MIMIC-III data This public dataset [10] has deidentified clinical care data collected at Beth Israel Deaconess Medical Center from 2001 to 2012. It contains over 58,000 hospital admission records of 38,645 adults and 7,875 neonates. For our work, we extracted 99 time series features from 19714 admission records for 4 events including input-events (fluids into patient, e.g., insulin), output-events (fluids out of the patient, e.g., urine), lab-events (lab test results, e.g., pH values and platelet count) and prescription-events (drugs prescribed by doctors, e.g., aspirin and potassium chloride). These events are known to be extremely useful for studying intensive care unit patients. All the time series are >48 hours of duration, and only the first 48 hours (after admission) time series data is used for training and testing our models. We perform following two predictive tasks on MIMIC-III.

- Mortality (MIMIC-III-Mor) task – Predict whether the patient dies in the hospital. There are 1716 patients with positive mortality label and we perform binary classification.
- ICD-9 Code Prediction (MIMIC-III-ICD9) task – Predict the ICD-9 diagnosis codes for each admission. There are 20 diagnoses[1] (e.g., respiratory system diagnosis) in our dataset. We treat it as a multi-task prediction problem.

3.2 Methods and implementation details

We categorize all evaluated methods into four groups:

1. *Non-RNN Baselines (Non-RNN)*: We evaluate logistic regression (LR), support vector machines (SVM) and Random Forest (RF) which are widely used models in health care applications.
2. *RNN Baselines (RNN)*: We consider LSTM-0, GRU-0, GRU-f and GRU-xmd from Section 2.3.
3. *Proposed Methods (Proposed)*: We test GRU-DM, GRU-DS and GRU-DI from Sections 2.4, 2.5.
4. *Ensemble Methods (Ensemble)*: We evaluate several ensembles of proposed and baselines methods.

The non-RNN baselines cannot handle missing data directly. We carefully design experiments for non-RNN models to capture the *informative missingness* as much as possible to have fair comparison with the RNN methods. Similar to RNN baselines, we can concatenate the mask vector along with the measurements and feed it to non-RNN models. However, the time duration vector cannot be concatenated since non-RNN models only work with fixed length inputs. We regularly sample the time-series data to get a fixed length input and perform imputation to fill in missing values. For PhysioNet dataset, we sample the time series on an hourly basis and propagate measurements forward (or backward) in time to fill gaps. For MIMIC-III dataset, we consider two hourly samples (in the first 48 hours) and do forward (or backward) imputation. Our preliminary experiments showed 2-hourly samples obtains better performance than one-hourly samples for MIMIC-III. We report results for both concatenation of input and mask vectors (e.g., SVM-xm, LR-xm, and RF-xm) and only input vector without mask (e.g., SVM-f, LR-f, and RF-f). We use the sklearn python library for the non-RNN model implementation and tune the parameters by cross-validation. For SVM, we choose RBF kernel since it performs better than other kernels.

For RNN models, we use a binary logistic regressor on top of the last hidden state $\boldsymbol{h}_T$ to do classification. We use 100 and 64 hidden units in GRU-0 for MIMIC-III and PhysioNet datasets,

[1] `http://www.tdrdata.com/ipd/ipd_SearchForICD9CodesAndDescriptions.aspx`

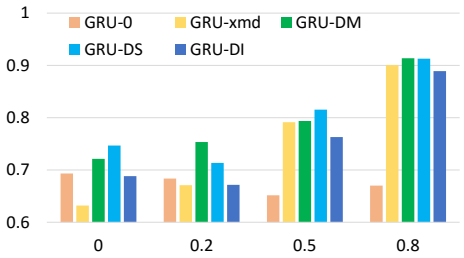


Figure 4: Performance on Gesture synthetic datasets with different correlations between missingness and labels. x-axis: average Pearson correlation score of the each variable's missing rate and the target label in that synthetic dataset; y-axis: AUC score.

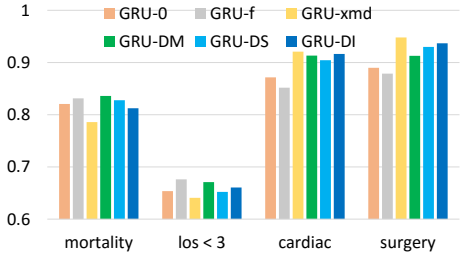


Figure 5: AUC score for 4 prediction tasks on PhysioNet dataset. x-axis: task; y-axis: AUC score.

respectively. All the other RNN models were constructed to have a comparable number of parameters. For GRU-xmd, we use mean imputation for input (Equation 1). We train all the RNN models with the Adam optimization method [11] and use early stopping to find the best weights on the validation dataset. All the input variables are normalized to be 0 mean and 1 standard deviation. We report the results from 5-fold cross validation for all the methods. For ensemble methods, we average the soft-labels of several classifiers and treat it as the ensemble prediction.

Recently RNN models have been explored for modeling diseases and patient diagnosis in health care domain [14, 5, 19] using doctor notes but are not readily applicable for comparison in our time series classification tasks since they don't handle missing data.

We will release our code to maximize reproducibility and to create a new benchmark for studying time series classification with missing data.

3.3 Quantitative results

Impact of missingness and label correlation on synthetic dataset To evaluate the impact of modeling missingness we conduct experiments on the synthetic Gesture datasets. Figure 4 shows the AUC score comparison of our proposed models (GRU-DM, GRU-DS, and GRU-DI) and two baseline GRU models (GRU-0 and GRU-xmd), given different correlations between missing rate and the label. Missing rate is the same for all the settings, but a higher correlation means the missingness is more informative. Since GRU-0 does not utilize masking or time duration, it performs similarly across all 4 correlation settings. All other models benefit from the missingness, especially when the correlation is high, and our proposed methods beat baselines in all settings. GRU-xmd, another baseline, performs well when correlation is high, but performs even poorer than GRU-0 when the correlation is low. This demonstrates that by simply concatenating the masking and time duration to the input, GRU-xmd cannot distinguish whether the missingness is useful or not. The results on synthetic datasets provide an insightful way to understand how our proposed models behave with different data properties.

Evaluation on real datasets Table 2 shows the prediction performance comparison of the models listed in Section 3.2 on mortality task for MIMIC-III and PhysioNet datasets. We observe the following: All models improve their performance when they feed missingness patterns along with inputs. Our proposed models achieve the best AUC score in both datasets. Our ensemble model based on proposed GRU models (GRU-DM, GRU-DS, and GRU-DI) and two non-RNN baseline models (SVM-xm and RF-xm) achieves the best performance with a significant improvement. This implies that our models exploit some knowledge which the baseline models do not capture. Also, Figure 5 and Figure 6 respectively show the AUC scores for all 20 ICD-9 diagnosis category prediction on MIMIC-III dataset and all 4 tasks on PhysioNet dataset for all RNN models. Our proposed models performs best in most of the tasks.

3.4 Discussions

Decay analysis Figure 7 shows the γ_t plots of all the variables for our GRU-DM model in Phy-Mor experiments. We observe that the decay rate is almost constant for variables that correspond to vital signs and a few lab measurements. Since these variables have less missing rate, our models do not

Table 2: Model performances measured by the Area Under ROC (AUC) score for predicting in-hospital mortality

Models		MIMIC-III	PhysioNet
Non-RNN	LR-f	0.7589 ± 0.015	0.7423 ± 0.011
	SVM-f	0.7908 ± 0.006	0.8131 ± 0.018
	RF-f	0.8293 ± 0.004	0.8183 ± 0.015
	LR-xm	0.7715 ± 0.015	0.7625 ± 0.004
	SVM-xm	0.8146 ± 0.008	0.8277 ± 0.012
	RF-xm	0.8294 ± 0.007	0.8157 ± 0.013
RNN	LSTM-0	0.8142 ± 0.014	0.8025 ± 0.013
	GRU-0	0.8066 ± 0.010	0.8087 ± 0.011
	GRU-f	0.8139 ± 0.008	0.8299 ± 0.011
	GRU-xmd	0.8371 ± 0.008	0.8215 ± 0.009
Proposed	GRU-DM	0.8411 ± 0.008	$\mathbf{0.8338 \pm 0.011}$
	GRU-DS	$\mathbf{0.8438 \pm 0.005}$	0.8229 ± 0.010
	GRU-DI	0.8421 ± 0.002	0.8202 ± 0.004
Ensemble		$\mathbf{0.8699 \pm 0.005}$	$\mathbf{0.8457 \pm 0.006}$

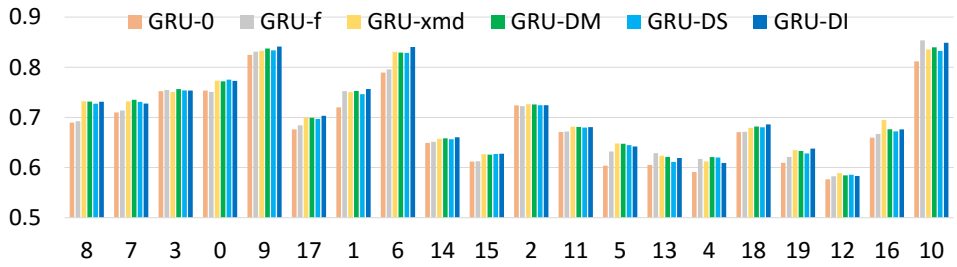


Figure 6: AUC score for 20 MIMIC-III ICD-9 diagnosis prediction tasks. x-axis: ICD-9 diagnosis category id; y-axis: AUC score. ICD-9 categories are ordered by the correlation of variable missing rate and labels; leftmost: ICD-9 category with highest correlation value; rightmost: lowest correlation value.

decay them over time. On the other hand, the variables with large decay mainly correspond to the lab test variables which have a long time duration between observations. Among these, variables such as Weight, Cholesterol, pH, Lactate, PaO2, etc. are known to be very important for clinical outcome prediction [29, 2] and thus, our model decays them appropriately so that their more recent observations are used for mortality prediction task.

Per time step prediction Although our model is trained on the prediction of last time step, it can be used directly to make predictions before it sees all the time series. This is very useful in applications such as health care, where early decision making is beneficial for patient care. Figure 8 shows the online prediction results for MIMIC-III mortality tasks, where the model makes prediction before it

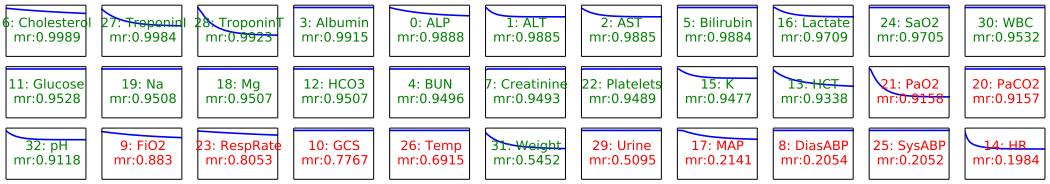


Figure 7: Decay γ_t plots of all 33 variables for Phy-Mor task in GRU-DM model. Variables in green: lab measurements; in red: vital signs. Variables are sorted in decreasing order of missing rate. mr: missing rate. x-axis: time duration δ_t^d, in range $[0, 24$ hours$]$; y-axis: decay value γ_t^d, in range $[0, 1]$.

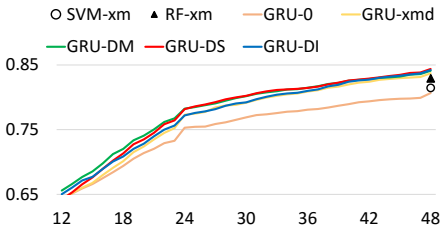


Figure 8: AUC score for online prediction on MIMIC-III mortality task. x-axis: # of hours after admission; y-axis: AUC score. SVM and RF results for 48 hours are shown in dash for reference.

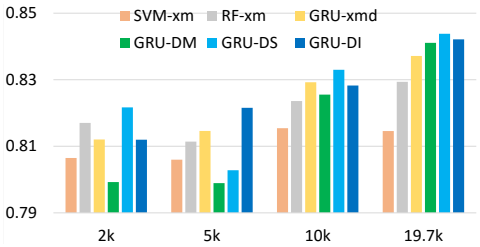


Figure 9: AUC on MIMIC-III mortality prediction with different number of training samples (2k, 5k, 10k, 19714). x-axis: training data size, y-axis: AUC score.

sees the entire time series. First, all the three proposed models beat the RNN baselines consistently from the very beginning. Second, with only part of the time series, the proposed methods can get the same or better performance than SVM and RF. Our models achieve similar prediction performance (i.e., same AUC) 11 hours earlier than SVM and 6 hours earlier than RF.

Performance with different training data size In many practical applications, model scalability with growing dataset size is very important. To evaluate the model performance with varying training dataset size, we generate three smaller datasets (2k, 5k, 10k admissions) from MIMIC-III by keeping the ratio of mortality label to dataset size similar to the original dataset. We compare our proposed models with three most competitive baseline models (SVM-xm, RF-xm, GRU-xmd) on these smaller datasets. We observe that all models can achieve improved performance from more training samples. However, the improvements of non-RNN baselines are quite limited compared with our models. This result indicates that the gap in performance between our models and the non-RNN baselines will continue to grow as more data becomes available.

4 Summary

In this paper, we proposed novel GRU-based models to effectively model multivariate time series with missing data. Our model captures the *informative missingness* by incorporating masking and time duration directly into the GRU architecture. Empirical experiments on real-world healthcare datasets showed promising results of our models. For future work, we are interested in exploring deep learning approaches to characterize missing-not-at-random data and conducting theoretical analysis to understand the behaviors of existing solutions to handle missing values.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] J. Bakker, M. W. Nijsten, and T. C. Jansen. Clinical use of lactate monitoring in critically ill patients. *Annals of intensive care*, 3(1):12, 2013.

[3] Y. Bengio and F. Gingras. Recurrent neural networks for missing or asynchronous data. *Advances in neural information processing systems*, pages 395–401, 1996.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] E. Choi, M. T. Bahadori, and J. Sun. Doctor ai: Predicting clinical events via recurrent neural networks. *arXiv preprint arXiv:1511.05942*, 2015.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010.

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] A. Johnson, T. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, and R. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 2016.

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2013.

[13] D. M. Kreindler and C. J. Lumsden. The effects of the irregular sample and missing data in time series analysis. *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data*, 2012.

[14] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzell. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[15] R. C. Madeo, C. A. Lima, and S. M. Peres. Gesture unit segmentation using support vector machines: segmenting gestures from rest positions. In *SAC*, 2013.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3, 2010.

[17] D. Mondal and D. B. Percival. Wavelet variance analysis for gappy time series. *Annals of the Institute of Statistical Mathematics*, 62(5):943–966, 2010.

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

[19] T. Pham, T. Tran, D. Phung, and S. Venkatesh. Deepcare: A deep dynamic memory model for predictive medicine. In *Advances in Knowledge Discovery and Data Mining*, pages 30–41. Springer, 2016.

[20] K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

[21] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[22] D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[23] J. L. Schafer and J. W. Graham. Missing data: our view of the state of the art. *Psychological methods*, 2002.

[24] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *CinC*, 2012.

[25] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[26] V. Tresp and T. Briegel. A solution for missing data in recurrent neural networks with an application to blood glucose prediction. *Advances in Neural Information Processing Systems*, pages 971–977, 1998.

[27] B. J. Wells, K. M. Chagin, A. S. Nowacki, and M. W. Kattan. Strategies for handling missing data in electronic health record derived data. *EGEMS*, 1(3), 2013.

[28] I. R. White, P. Royston, and A. M. Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.

[29] J.-W. You, S. J. Lee, Y. E. Kim, Y. J. Cho, Y. Y. Jeong, H. C. Kim, J. D. Lee, J. R. Kim, and Y. S. Hwang. Association between weight change and clinical outcomes in critically ill patients. *Journal of critical care*, 28(6):923–927, 2013.

A Supplementary

A.1 MIMIC-III preprocessing details

Here, we describe the preprocessing details for MIMIC-III dataset[2]. MIMIC-III provides several relational database tables containing information of data relating to patients who stayed within the intensive care units (ICUs) at Beth Israel Deaconess Medical Center. The admission table contains over 58,000 hospital admission records of 38,645 adults and 7,875 neonates. We chose four tables namely inputevents-mv (fluids into patient, e.g. insulin), outputevents (fluids out of the patient, e.g. urine), labevents (lab test results, e,g. pH, Platelet count) and prescription events (drugs prescribed by doctors, e.g. aspirin and potassium chloride) to collect the patient data recorded in critical care units and hospital record systems. The inputevents-mv table collects the intake for patients monitored using the iMDSoft Metavision system. For our work, we use 19714 admission records collected during 2008-2012 by Metavision data management system which is still employed at the hospital. The data collection and organization in Metavision system is much neater than the earlier Philips CareVue system [2001-2008]. From each of the four tables, we chose the top 50 items (i.e. features/variables) since these items are present in many of the patients' records. To avoid/reduce ambiguity and noisy observations, we ensured that all the measurements for a particular variable has only one unit of measurement. We also aggregated the multiple readings of a feature at a single time stamp based on the feature type. For instance, some inputevents features should be averaged while others need to be summed up. This resulted in 99 variables being extracted from the four tables for 19714 patient admission records. For each of the admission records, we collected both the variable value x_t and the time-stamp of observation s_t. In addition, for each admission record we queried the database tables to get the ICD-9 diagnosis codes. One admission record can be associated with multiple ICD-9 codes. We also queried the discharge time and death time from the Admissions table of MIMIC-III to find the mortality label for each admission record. The ICD-9 diagnosis codes, shown in Table 3 were grouped into 20 categories according to the information from the Thomson Reuters webpage[3]. The class distribution of the ICD-9 codes is shown in Figure 10.

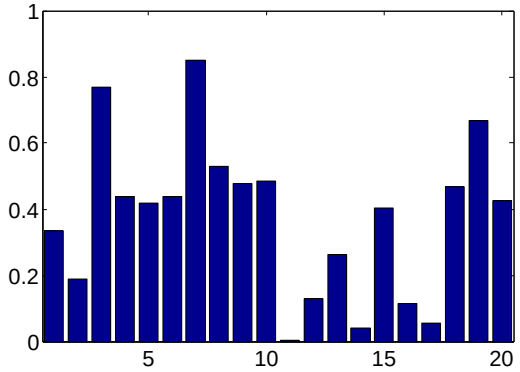


Figure 10: MIMIC-III ICD-9 diagnosis code class distribution. x-axis, ICD-9 diagnosis category id; y-axis: the ratio of admission records with the diagnosis code.

A.2 Descriptions for Figure 1

In many time series applications, the pattern of missing variables in the time series is often informative and useful for prediction tasks. Here, we empirically confirm this claim on two health care datasets by investigating the correlation between the missingness and prediction label (mortality prediction task). We compute the Pearson correlation coefficient between $p^d_{\boldsymbol{X}}$ and label ℓ across the training time series. As shown in Figure 1, we observe that the variables with low missing rate are highly correlated with the target label, demonstrating the usefulness of missingness patterns in solving a prediction task. Note that $p^d_{\boldsymbol{X}}$ is dependent on mask vector (m^d_t) and number of time steps T.

[2] `https://mimic.physionet.org/`

[3] `http://tdrdata.com/ipd/ipd_SearchForICD9CodesAndDescriptions.aspx`

Table 3: MIMIC-III ICD-9 diagnoses tasks description

Task ID	ICD-9 Codes	Diagnoses Groups
1	001 - 139	Infectious and Parasitic Diseases
2	140 - 239	Neoplasms
3	240 - 279	Endocrine, Nutritional, Metabolic, Immunity
4	280 - 289	Blood and Blood-Forming Organs
5	290 - 319	Mental Disorders
6	320 - 389	Nervous System and Sense Organs
7	390 - 459	Circulatory System
8	460 - 519	Respiratory System
9	520 - 579	Digestive System
10	580 - 629	Genitourinary System
11	630 - 677	Pregnancy, Childbirth, and the Puerperium
12	680 - 709	Skin and Subcutaneous Tissue
13	710 - 739	Musculoskeletal System and Connective Tissue
14	740 - 759	Congenital Anomalies
15	780 - 789	Symptoms
16	790 - 796	Nonspecific Abnormal Findings
17	797 - 799	Ill-defined and Unknown Causes of Morbidity and Mortality
18	800 - 999	Injury and Poisoning
19	V Codes	Supplemental V-Codes
20	E Codes	Supplemental E-Codes