# Phased-LSTM Based Predictive Model for longitudinal EHR Data with Missing Values

**Seo-Jin Bang** SEOJINB@ANDREW.CMU.EDU
**Yuchuan Wang** YUCHUANW@ANDREW.CMU.EDU
**Yang Yang** YY3@ANDREW.CMU.EDU

## 1. Introduction

Electronic health records (EHRs) is an inter organizational, comprehensive, patient-centered longitudinal collection of health records. It aims to foster the quality of health care and support healthcare providers by providing comprehensive medical and healthcare information of patients along the whole life cycle. In particular, EHRs can be utilized for diagnosis of current status of a patient such as existence of specific diseases or prediction of future status of a patient such as survival probability after three years of a surgery.

Previous researches (Schulam & Saria, 2016; Wang et al., 2012; Rizopoulos & Ghosh, 2011) have worked on extracting meaningful patterns from the EHRs while most of them failed to capture irregular patterns of sampling or long range dependencies from the multivariate, varying length time-series record of observations. A somewhat successful research (Lipton et al., 2015) use Long Short-Term Memory (LSTM) to construct a diagnosis model that effectively captures time-series observations with variation of the length and long range dependencies, while it could not capture *irregularity of sampling interval*. Note that the irregularity of sampling interval means that intervals between two consecutive visits have varying lengths (See Figure 3-(a)). In order to deal with the irregular pattern of sampling, Neil et al. (2016) suggested Phased-LSTM to improve current recurrent neural network (RNN) models that are ill-suited by adding a new time gate to deal with irregularly sampled time-series data generated in continuous time while it was not designed as a biologically plausible model. However, it is not straightforward to apply the Phased-LSTM to EHRs because each clinical features in EHRs are asynchronously sampled. The two missing patterns induced from irregularly sampled subjects and asynchronously sampled features complicate the missing mechanism of EHRs (See Figure 3). In the past decades, several researches have been developed to address the problem of asynchronously sampled features. Most approaches suggested two-step processes applying missing imputation techniques to fill asynchronously sampled features first and then learning prediction models on the completed data

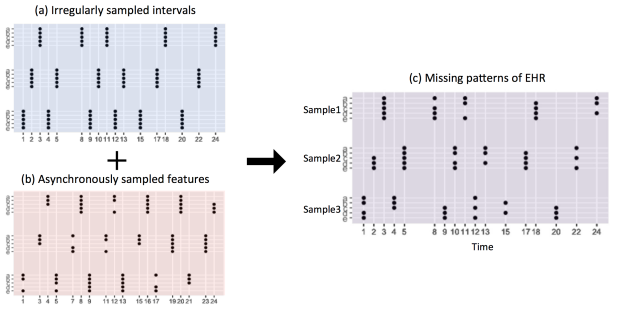



*Figure 1.* (a) Irregularly sampled intervals: each of the three samples is irregularly sampled in continuous time while five features $(a, b, c, d, e)$ are simultaneously collected at each time point. (b) Asynchronously sampled features: the features are asynchronously sampled within each time while each time point is regularly spaced. (c) Missing pattern of EHR is composed of both the two missing patterns (a) and (b).

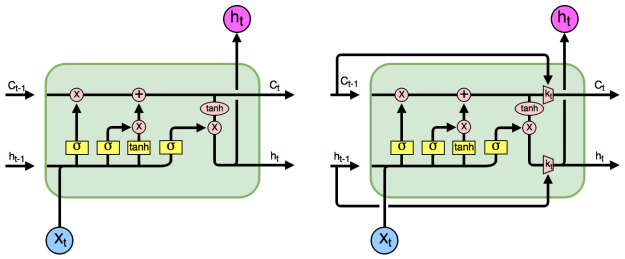
(Kreindler & Lumsden; Rehfeld et al., 2011; White et al., 2011; García-Laencina et al., 2010; Lipton et al., 2016). However, the asynchronism of clinical features is resulted from experimental design rather than the assumption of complete/incomplete randomness made from most of the missing imputation techniques. Recently, Che et al. (2016) suggested a deep learning model, GRU-D, based on Gated Recurrent Unit (GRU, Cho et al. (2014)) to capture the long-term temporal dependencies in time series as well as model the missing patterns. In particular, GRU-D models decay mechanism of inputs and the hidden states using two representations of the missing patterns: masking vector and time interval. However, it considers the two different missing patterns, induced from irregular sampling among subjects and asynchronous sampling among features within a subject, as an unified missing mechanism. It results in discarding useful information about when each observation is recorded since GRU-D imputes values of the features at all time points even when the patient did not visit.

We suggest approaches to learn predictive deep learning models using Phased-LSTM from longitudinal EHRs which can be used for disease diagnosis and prediction. First four approaches use the existing model, Phased-LSTM, which is suggested to address the problem of ir-

regular sampling, while it have not been applied to EHR data. In order to deal with the problem of missing values resulted from the asynchronous feature sampling, we impute the missing features first using various missing imputation approaches described in Sec 3.2.1 in detail. This approach includes Phased-LSTM-Forward, Phased-LSTM-Linear, Phased-LSTM-Masking and Phased-LSTM-ALL, which are named based on the imputation method used to replace the missing values. Second, we suggest a new model *Phased-LSTM-D* that embeds the decay rate $\gamma_t$ suggested by Che et al. (2016) into Phased-LSTM. Phased-LSTM-D takes advantages of both Phased-LSTM and GRU-D. The cell state $c_t$ allows information to pass the LSTM cell so it can capture long range dependencies. The new time gate $k_t$ makes to efficiently learn from the irregularly sampled time-series data generated in continuous time as it does in Phased-LSTM. In order to deal with asynchronously sampled features, it also introduces the decay rate $\gamma_t$ to impute inputs and update hidden states as it does in GRU-D. We applied the proposed models to the Physionet Challenge 2012 dataset (Silva et al., 2012) for making predictions of in-hospital death of the patients based on the EHR data, in comparison with four methods based on the Phased-LSTM model and in-advance feature imputation or reformulation strategies. The model is evaluated by predicted mortality status with the true status using different evaluation metrics.

## 2. Background and Related Work

Recurrent Neural Network (RNN) is a class of artificial neural network with a chain-like structure of repeating modules of neural network connecting the hidden unit to form a sequence. The repeating module of standard RNN has a simple single hidden layer such as tanh layer. The hidden units capture the information of what happened in previous time steps. Hence, it has become a state-of-the-art choice for extracting patterns from longitudinal sequences. However, in training of the standard RNN, it is not capable of handling long-term dependencies in which the temporal contingencies present in and input/output sequences span long intervals, while in theory it is (Bengio et al., 1994). Long Short-Term Memory (LSTM, Hochreiter & Schmidhuber (1997)) is a specific kind of RNN, which contains LSTM units as the hidden unit. It has been used to catch the long-term dependencies by remembering information for long periods of time. The main difference to standard RNN is that LSTM has a cell state $c_t$ at time $t$ itself updated with a fraction of previous cell state to convey the long range dependencies (See Figure 2-(a)). It has three gates to protect and control the cell state $c_t$: input gate $i_t$, forget gate $f_t$, and output gate $o_t$. Lipton et al. (2015) used LSTMs to recognize patterns from an EHR dataset in which each observation consists of 13 frequently but irregularly sampled




(a) Standard LSTM (b) Phased LSTM

*Figure 2.* Model architecture. (a) Standard LSTM model: LSTM has a cell state $c_t$ allows long range information to persist along with the hidden states. We borrows the visualization style of the architecture from Olah (2015). (b) Phased-LSTM model: Phased-LSTM has additional time gate $k_t$ to turn on and off the update of the cell state $c_t$ and hidden output state $h_t$, which can model the irregular sampling intervals and correctly sample at any continuous time.

time series of clinical measurements including body temperature, heart rate, diastolic and systolic blood pressure, blood glucose, etc. It associates each patient with a subset of the 128 most common diagnosis codes, classifying each episode with one or more diagnoses. They showed that LSTM can effectively model long range dependencies and nonlinear dynamics of the EHR data. However, under the irregular sampling such that intervals between two consecutive inputs are not evenly spaced and are instead sampled at irregular times, LSTM cannot be effectively learned from the data.

In order to deal with the problem, Neil et al. (2016) suggested Phased-LSTM, which extends the LSTM unit by adding a new time gate $k_t$ (See Figure 2-(b)). It showed that Phased-LSTM has no difficulty with the irregularly sampled data, achieving higher accuracy compared to traditional LSTM. At each time step $t$, a parameterized oscillation turns the gate $k_t$ on and off to update the cell state and the hidden output state. The updates can only happen during an *open* phase, otherwise the previous parameters will be kept. The time gate models the irregular updates and can be correctly sampled at any continuous time. Phased-LSTM also out-performs traditional LSTM in terms of convergence and running time. The authors applied Phased-LSTM to several tasks like frequency discrimination, visual recognition and lip-reading. Even with the sparse updates of the memory cell controlled by the oscillation of the time gate, the Phased-LSTM converges more quickly and requires only $5\%$ of the computes at runtime, while often improving in accuracy compared to standard LSTM. Since the nature of EHR data is event-based, there is potential for Phased-LSTM to model such data.

However, in practice, samples are not only irregularly measured in continuous time but also not all clinical features are measured at the same time (See Figure 3). Combi-

nation of the two missing patterns makes EHR data hard to be learned. Most approaches to deal with the problem of asynchronously measured time-series features are composed of two-steps, explicitly applying missing imputation techniques to fill the asynchronously sampled features and then fitting a prediction model on it (Kreindler & Lumsden; Rehfeld et al., 2011; White et al., 2011; García-Laencina et al., 2010; Lipton et al., 2016). For example, Kreindler & Lumsden imputed missing data segments by means of segment concatenation using segment filling with average data values or local interpolation in phase space. White et al. (2011) suggested MICE utilizing multiple imputation approach using chained equations. The posterior distribution of a variable to be imputed is approximated by regression dependent on all the other remaining variables. García-Laencina et al. (2010) assumed that missing patterns are extracted from a probability distribution and made used of EM algorithm to impute missing values. Lipton et al. (2016) imputed missing values using forward- and back-filling imputation strategies and then learn LSTM for multi-label classification tasks from clinical time series data.

However, the missing imputation approaches discard useful information about when each observation is recorded. Moreover, the asynchronism of feature sampling is typically natural outcomes of the experimental design which is far from the assumption of complete/incomplete randomness made from most of the missing imputation techniques.

Recently, Che et al. (2016) suggested GRU-D that models decay mechanism of inputs and hidden output states by adding decay rates $\gamma_t$ to standard Gated Recurrent Units (GRU, Cho et al. (2014)). GRU-D assumes two principles behind the decay mechanism of EHR data: (1) the missing values of features approach to some default value if its last observation happens a long time ago and (2) influence of previous inputs will fade away as time goes. The decay rate $\gamma_t$ is simultaneously trained with the model and used to impute missing values and update hidden states. While it mentioned that the decay term also can be embedded into LSTM straightforwardly, it has not been implemented to LSTM. Moreover, it counts the two missing mechanisms, the irregularity of sampling intervals and the asynchronism of sampling features, without discrimination, while the first mechanism is resulted from visiting cycle of patients and the second mechanism is resulted from asynchronism of clinical measurements.

## 3. Methods

We learn predictive models from EHR data using two different approaches. First, we employ the existing model, Phased-LSTM, which is suggested to address the problem of irregular sampling, while it has not been applied to the EHR data. Before training Phased-LSTM on EHR data, we imputed the missing features first using various missing imputation approaches described in Sec 3.2.1 in detail. The imputation aims to deal with the problem of missing values resulted from the asynchronous feature sampling. This approach includes Phased-LSTM-Forward, Phased-LSTM-Linear, Phased-LSTM-Masking and Phased-LSTM-ALL, which are named based on the imputation method used to replace the missing values.

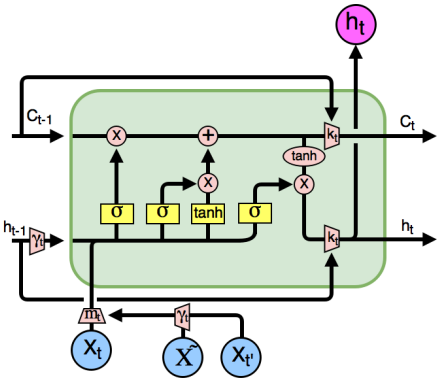



*Figure 3.* Model architecture of Phased-LSTM-D

Second, we suggest a new predictive deep learning model, *Phased-LSTM-D*, that can simultaneously but separately deal with the two missing patterns: (1) irregularity of sampling intervals and (2) asynchronism of sampling features. It utilizes the decay rate $\gamma_t$ suggested by Che et al. (2016) to deal with the asynchronous feature sampling problem. Phased-LSTM-D takes advantages of both Phased-LSTM and GRU-D by embedding the decay rate $\gamma_t$ into Phased-LSTM. Figure 3 illustrates the model architecture of Phased-LSTM-D. Like Phased-LSTM, the cell state $c_t$ conveys information along with the sequence of hidden units, which allows to model long range dependencies. The input gate $i_t$ and forget gate $f_t$ control the update of the cell state $c_t$. The time gate $k_t$ controls the cell state $c_t$ and hidden output state $h_t$ to be updated only at event-driven time points, while allows some levels of information leaked. Meanwhile, Phased-LSTM-D models decay mechanism of inputs and hidden states using decay rate $\gamma_x$ and $\gamma_h$ respectively at each event-driven time point $t_j$ for $j = 1, \cdots, T_n$. When the inputs are missing at time $t_j$, the missing features $x_j$ induced from the asynchronously sampled features are replaced with a weighted sum of the last measurement $x_{p(t_j)}$ and average measurement $\tilde{x}$ controlled by a decay rate $\gamma_j$. The previous hidden state $h_{j-1}$ is also updated with a fraction of previous hidden state controlled by the decay rate $\gamma_j$.

### 3.1. Data Processing

EHR data has two different missing patterns: the irregularity of sampling intervals and the asynchrony of features. In order to deal with the missing patterns, we pre-processed

the data before learning predictive model from longitudinal EHR.

Suppose there are $N$ patients, $D$ time-series features and $T$ time points in total. For simplicity we first only consider the record of the $n$-th patient. Each $n$-th patient has five types variables such as:

$$\begin{aligned} X_n &= (x_1, \cdots, x_{T_n})^\top \in \mathbb{R}^{T_n \times D} \\ S_n &= (s_1, \cdots, s_{T_n}) \in \mathbb{R}^{T_n} \\ M_n &= (m_1, \cdots, m_{T_n})^\top \in \mathbb{R}^{T_n \times D} \\ \Delta_n &= (\delta_1, \cdots, \delta_{T_n})^\top \in \mathbb{R}^{T_n \times D} \\ X_n^{prev} &= (x_1, x_{p(2)}, \cdots, x_{p(T_n)})^\top \in \mathbb{R}^{T_n \times D} \end{aligned}$$

and one global vector $X_n^{mean} = (\tilde{x}_n^{(1)}, \cdots, \tilde{x}_n^{(D)}) \in \mathbb{R}^D$.

The time times feature vector is $X = (x_1, \cdots, x_{T_n})^\top \in \mathbb{R}^{T_n \times D}$, where $T_n$ is the number of time points when the $n$-th patient is sampled, and $x_{t_j}$ for $j = 1, \cdots, T_n$ represents values of the $D$ features at time step $t_j$. In case of the vector $x_{t_j}$ has missing features. We mark the missing features with a binary masking vector $m_{t_j}^{(d)}$ to indicate whether a feature $d$ is measured or not at time step $t_j$ such that:

$$m_{t_j}^{(d)} = \begin{cases} 1, & \text{if } x_{t_j}^{(d)} \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

We use $s = (s_1, \cdots, s_{T_n})$ to represent all the time points when $n$-th patient is sampled. We define $\delta_{t_j}$ as a $D$-dimensional time interval vector, of which $\delta_{t_j}^{(d)}$ represents the time interval between the current time step $t_j$ and the time step when the last observation of $x_{t_j}^{(d)}$ is available as follow:

$$\delta_{t_j}^{(d)} = \begin{cases} s_j - s_{j-1} + \delta_{j-1}^{(d)}, & \text{if } j > 1, m_{j-1}^{(d)} = 0 \\ s_j - s_{j-1}, & \text{if } j > 1, m_{j-1}^{(d)} = 1 \\ 0, & \text{if j=1,} \end{cases} \tag{2}$$

We also define the empirical mean $\tilde{x}_n^{(d)}$ of the $d$-th variable $x_n^{(d)}$ for the $n$th patient over all the time points as follow:

$$\tilde{x}_n^{(d)} = \frac{\sum_{j=1}^{T_n} m_{n,t_j}^{(d)} x_{n,t_j}^{(d)}}{\sum_{j=1}^{T_n} m_{n,t_j}^{(d)}} \tag{3}$$

The last available observation for each $x_{t_j}^{(d)}$ is also denoted as $x_{p_d(t_j)}^{(d)}$ where $p_d(t_j) = \max\{t' \mid t' < t_j$, the feature $d$ is observed at $t'\}$.

### 3.2. Predictive Model using Phased-LSTM

#### 3.2.1. Missing Value Imputation

Before we learn Phased-LSTM which is able to resolve the problem of the irregularly sampled intervals, we employ several feature imputation or reformulation approaches to address the problem of asynchronously sampled features which results in missing values of the features at each time point.

- The first approach is to simply use the last available observation $x_{p_d(t)}^{(d)}$ to impute the missing value, i.e., passing valid observations forward along the time to replace missing feature values.

- The second approach is the perform linear interpolation to estimate the missing value. Suppose the $d$-th variable $x_{t_j}^{(d)}$ is missing at time step $t_j$. We estimate $x_{t_j}^{(d)}$ using the last available observation $x_{p(t_j)}^{(d)}$ and the next available observation $x_{q(t_j)}^{(d)}$ as follow:

  $$x_{t_j}^{(d)} \leftarrow \gamma x_{p(t_j)}^{(d)} + (1-\gamma) x_{q(t_j)}^{(d)},$$

  where $\gamma = (s_{q(t_j)} - s_{t_j})/(s_{q(t_j)} - s_{p(t_j)})$, $s_{t_j}$ is the time point at step $t_j$.

- The third approach (Che et al., 2016) does not impute the missing values directly. It instead concatenates the original feature vector $x_{t_j}$ with the masking vector $m_{t_j}$ and the time interval vector $\delta_{t_j}$ to formulate a new feature vector at each time step, such that

  $$\hat{x}_{t_j} \leftarrow [x_{t_j};\ m_{t_j};\ \delta_{t_j}], \tag{4}$$

  where $\hat{x}_{t_j} \in \mathbb{R}^{3 \times D}$. In this way we incorporate the information of which features are missing and how long they have been missing into reformulation of a new feature vector.

- The fourth approach is to only use the masking vector $m_t$ in combination with $x_{t_j}$ to formulate the new feature vector without using $\delta_{t_j}$.

Each of the four approaches can be used in combination with the Phased-LSTM. We name the four combined methods as Phased-LSTM-Forward (using the last observation for data imputation), Phased-LSTM-Linear, Phased-LSTM-ALL and Phased-LSTM-Masking, respectively.

#### 3.2.2. Learn Phased-LSTM from Imputed EHR

We investigate the use of Phased-LSTM for diagnosis and prediction of disease status from EHR with the irregularity of sampling intervals. Note that the missing values induced from the asynchronously sampled feature are already imputed, and we learn Phase-LSTM from the imputed EHR.

First, we start with the standard LSTM:

$$i_t = \sigma_i(x_t W_{xi} + h_{t-1} W_{hi} + b_i) \quad (5)$$
$$f_t = \sigma_f(x_t W_{xf} + h_{t-1} W_{hf} + b_f) \quad (6)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \quad (7)$$
$$o_t = \sigma_o(x_t W_{xo} + h_{t-1} W_{ho} + b_o) \quad (8)$$
$$h_t = o_t \odot \sigma_h(c_t) \quad (9)$$

The main improvement of LSTM compared to the standard RNN is to use separated state, called cell state $c_t$ passing informations through time sequence, from the hidden output state $h_t$ (See Figure 2-(a)). The use of the cell state allows to keep long range dependencies between inputs and outputs. The cell state $c_t$ is updated with a fraction of previous cell state $c_{t-1}$ controlled by the *forget* gate $f_t$ and a fraction of a non-linear function of the input $x_t$ and previous hidden output state $h_{t-1}$ controlled by the *input* gate $i_t$. The hidden output state $h_t$ is a fraction of a non-linear function of current cell state $c_t$ controlled by *output* gate $o_t$. The input, forget, and output gate, $i_t$, $f_t$, and $o_t$, use typical sigmoid functions $\sigma_i$, $\sigma_f$, and $\sigma_o$. The tanh nonlinearities $\sigma_c$ and $\sigma_h$ with weight parameters $W_{hi}$, $W_{hf}$, $W_{ho}$, $W_{xi}$, $W_{xf}$, and $W_{xo}$ are used to connect different inputs and gates with the memory cells and outputs.

Note that in a typical LSTM model $c_t$ and $h_t$ are regularly updated at every time points, while EHR is collected with irregularly spaced time interval. In order to model the irregularity of sampling intervals, Phased-LSTM (Neil et al., 2016) adds a new time gate $k_t$ (See Figure 2-(b)) to achieve more efficient training over the irregularly sampled data. The cell state $c_t$ and $h_t$ can be updated only when $k_t$ is open. In this way Phased-LSTM can address the problem of a typical LSTM that the memory decays exponentially with the time steps. The time gate $k_t$ are controlled by a rhythmic oscillation modeled by three parameters, $\tau$, $r_{on}$ and $s$: $\tau$ represents the real-time period of the oscillation, $r_{on}$ controls the ratio of of the open phase duration $k_t$ to be opened over the whole period, and $s$ represents the shifting phase of oscillation to each cell. The typical definition of the time gate $k_t$ suggested by (Neil et al., 2016) is:

$$k_t = \begin{cases} \frac{2\phi_t}{r_{on}}, & \text{if } \phi_t < \frac{1}{2} r_{on} \\ 2 - \frac{2\phi_t}{r_{on}}, & \text{if } \frac{1}{2} r_{on} < \phi_t < r_{on} \\ \alpha \phi_t, & \text{otherwise} \end{cases}$$

where $\phi_t$ represents the phase inside the rhymic cycle defined as:

$$\phi_t = \frac{(t-s) \text{mod} \tau}{\tau}.$$

Openness of the time gate $k_t$ rises from 0 to 1 during the first phase and drops from 1 to 0. During the third phase, the time gate $k_t$ is closed so that the previous cell state is mostly maintained while the leaky parameter $\alpha$ propagate important gradient information.

From here we use a new notation such that $j$ represents the update time point $t_j$ and $j-1$ represents the previous update time point $t_{j-1}$. The cell state $c_j$ and the hidden output state $h_j$, are updated only when the sample is observed at the time point $t_j$. Therefore, Phased-LSTM sparsely updates its status at the irregularly sampled time points instead of being updated at every time step, which is controlled by the time gate $k_j$. In detail, instead of following the updating rule described in Eq 7, the cell state $c_j$ is updated as follow:

$$\widetilde{c}_j = f_j \odot c_{j-1} + i_j \odot \sigma_c(x_j W_{xc} + h_{j-1} W_{hc} + b_c)$$
$$c_j = k_j \odot \widetilde{c}_j + (1 - k_j) \odot c_{j-1}.$$

Similarly, instead of following the updating rule Eq 9, the hidden output state $h_j$ is updated as follow:

$$\widetilde{h}_j = o_j \odot \sigma_h(\widetilde{c}_j)$$
$$h_j = k_j \odot \widetilde{h}_j + (1 - k_j) \odot h_{j-1}.$$

Phased-LSTM models rate of memory decay using the time gate $k_t$ controlled by independent rhythmic oscillation. Therefore, the gradient back-propagates through fewer updating time steps leads to slower decay of the memory and possibly faster convergence of the learning process.

### 3.3. Predictive Model using Phased-LSTM-D

Note that the imputation for missing values of asynchronously sampled features should precede the learning of Phased-LSTM on EHR data. Now we suggest *Phased-LSTM-D* to simultaneously deal with the asynchronously sampled features in addition to the irregularity of sampling intervals. Phased-LSTM-D embeds the decay mechanism for the asynchronously sampled features suggested by Cho et al. (2014) into Phased-LSTM to capture the long-term temporal dependencies in continuous time as well as separately model the two different missing patterns. Note that we assumes that there is two different missing mechanisms: (1) irregularity of sampling intervals and (2) asynchronism of sampling features. To deal with the irregularity of sampling interval, we adapt basic framework of Phased-LSTM using the time gate $k_t$ controlling the gate for updating the cell state $c_t$ and the hidden output state $h_t$ on and off. Note that the update of the Phased-LSTM is event-based: the states of a patient $n$ are updated only at each time sequences $t_j$ for $j = 1, \cdots, T_n$ where the patient $n$ is observed. That is, the time gate $k_t$ only opens at $t = t_1, \cdots, t_{T_n}$. At each event $t_j$, the missing features $x_j$ induced from the asynchronously sampled features are replaced with a weighted sum of the last measurement $x_{j'}$ and average measurement $\tilde{x}$ controlled by a decay rate $\gamma_j$. The previous hidden state $h_{j-1}$ is also replaced with a fraction of previous hidden state controlled by the decay rate $\gamma_j$.

I. OPEN PERIOD

When the time gate is open, the missing feature $x_j^{(d)}$ are replaced as follow:

$$x_j^{(d)} \leftarrow m_j^{(d)} x_j^{(d)} + (1 - m_j^{(d)})\gamma_{x_j}^{(d)} x_{j'}^{(d)} + (1 - m_j^{(d)})(1 - \gamma_{x_j}^{(d)})\tilde{x}^{(d)}.$$

Note that we use same definitions and notations as we defined in Sec 3.1. That is, $x_j^{(d)}$ be the value of the $d$-th variable at time $t_j$; $\tilde{x}^d$ denotes the empirical mean of the $d$-th variable; and $x_{j'}^{(d)}$ be the last available observation of $x^{(d)}$ (i.e. $t_{j'} < t_j$). The masking variable $m_j^{(d)}$ (See Eq (1) for definition) tells whether the feature $d$ is missing or not at the event $t_j$ so it indicates whether the missing value will be replaced with the weighted sum of $\tilde{x}^d$ and $x_{j'}^{(d)}$ or the observed value $x_j^{(d)}$ will be used.

The decay rate $\gamma_{x_j}^{(d)}$ for the input feature $d$ at time $t_j$ controls the weight between $\tilde{x}^{(d)}$ and $x_{j'}^{(d)}$. The $D$-dimensional vector $\gamma_{x_j} = \left[\gamma_{x_j}^{(1)}, \cdots, \gamma_{x_j}^{(D)}\right]$ is defined as:

$$\gamma_{x_j} = \exp\{-\max(0, W_{\gamma_x}\delta_j + b_{\gamma_x})\} \tag{10}$$

where $W_\gamma$ and $b_\gamma$ are model parameters to be trained jointly with the other parameters. $\delta_j^{(d)}$ (See Eq (2) for definition) represents the time interval between the current time step and the time step when the last observation of $x_j^{(d)}$ is available.

Similarly, we define a decay rate for the previous hidden state $h_{j-1}$ at time $j$,

$$\gamma_{h_j} = \exp\{-\max(0, W_{\gamma_h}\delta_j + b_{\gamma_h})\} \tag{11}$$

The decay rate $\gamma_{h_j}^{(d)}$ also controls a fraction of previous hidden state $h_{j-1}$ to be used as an input for the current cell.

$$h_{j-1} \leftarrow \gamma_{h_j} \odot h_{j-1}. \tag{12}$$

That is, the previous hidden state $h_{j-1}$ is decayed before it is used to compute the next hidden output state $h_j$ as follow:

II. CLOSED PERIOD

When all the features of a sample $n$ are missing at time $t$, the gate $k_t$ will be closed, which the missing pattern is induced from the irregular pattern of subject sampling. Phased-LSTM-D simply works exactly same as Phased-LSTM as described in Sec 3.2.

# 4. Experiments

## 4.1. Data collection

The dataset we used for this study is the Physionet Challenge 2012 dataset (Silva et al., 2012). This dataset contains time-series multivariate intensive care unit (ICU) records of the first 48 hours after a patient was admitted to the ICU. The Training Set A comprises 4000 records with available label information of mortality. The records contain 6 general descriptors, and 36 time-series variables in total, such as Albumin, Hematocrit and Heart rate. The number of measured time-series variables varies in different records, with the average around 26. There are six outcome-related descriptors. They are respectively recordID, SAPS-I score(cite), SOFA score(cite), length of stay (days), survival (days) and in-hospital death (0: survivor or 1: died in hospital).

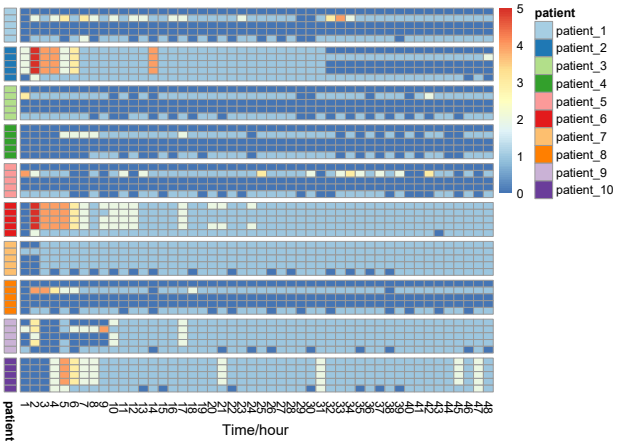



*Figure 4.* Physionet Challenge 2012 ICU records of ten patients randomly selected. X-axis represents the time. Y-axis represents five features: HR, SysABP, DiasABP, MAP, and Urine.

We observed both the irregular sampling problem and asynchronism of feature sampling problem in this dataset (See Fig 4). The 36 time series variables were measured at varied frequencies across the records. Some variables (such as heart rate and blood pressure) were measured much more frequently than the others, while some variables were measured only few times. The time points where measurements were taken within the 48 hours are not only irregularly distributed within a single record, but also change among different records. Among the 2881 time points within the 48 hours (unit:minute), the number of sampled time points for each patient ranges from 50 to 120, and all the 2881 time points were sampled for at least one patient. We present the irregularity and asynchronism of feature sampling for 10 randomly chosen records, as shown in Figure 4.

In our experiment, we aim to use the multivariate time-series ICU record data to predict the in-hospital death of

a patient. We need to address the problems of the irregular sampling and missing features due to the asynchronous feature sampling.

### 4.2. Result

First we extracted all the six types of variables for each record from the Physionet Challenge 2012 dataset, as described in Sec 3.1. They are respectively $X_n$ (for the $n$-th record), $S_n$, $M_n$, $\Delta_n$, $X_n^{prev}$ and $X_n^{mean}$, $n = 1, \cdots N$. The label is a binary variable indicating the in-hospital death of the corresponding patient. There are respectively 554 positive samples (in-hospital death: 1) and 3446 negative samples (in-hospital death: 0). We divided the samples into training data and test data based on 5-fold cross validation. Thus for each fold, there are 3200 training samples and 800 test samples.

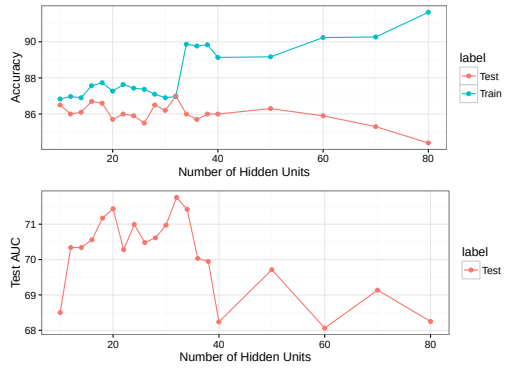



*Figure 5.* Accuracy and AUC versus the number of hidden units

We applied all the five methods as described in Sec 3.2 to the dataset for model training and making predictions for the test samples. The first four methods are Phased-LSTM-Forward, Phased-LSTM-Linear, Phased-LSTM-All and Phased-LSTM-Masking respectively. The fifth method is the proposed Phased-LSTM-D.

The feature vector of each sample varies upon the specific method applied. The first four methods are all based on the Phased-LSTM model. For the Phased-LSTM-D model, We use additional information such as feature masking vectors and time interval vectors to train additional hidden layers in the proposed neural network for estimating the decay coefficients, in order to perform input decay and hidden state decay simultaneously with the model training.

We used 60 epochs and the batch size of 50 for each of the methods in model training. Performance evaluation is based on 5-fold cross validation for each of the five methods. All the five methods use the same set of 5-fold splits. The evaluation metrics we employ are accuracy, AUC (area under the Receive Operating Characteristic curve) and the loss function value. Accuracy is defined as $(TP + TN)/(P + N)$. The loss function is computed as the cross entropy between the predicted labels and the ground truth labels. For each method, the average of prediction performance on each fold with respect to each metric is used for evaluation. To determine the appropriate number of hidden units in the model, we ran Phased-LSTM-D on Physionet Challenge 2012 dataset with different number of hidden units (ranging from 10 to 80). For different number of hidden units, we compared the prediction accuracy for training and testing data to see if the model is over-fitted. We found that when the number of hidden units is between 10 and 32, the prediction accuracy for training data is slightly higher than testing data. But if the number of hidden units is larger than 32, the prediction accuracy of training data will be much higher than testing data and testing AUC will start to decrease (Figure 5), suggesting that the model is over-fitted. Therefore, we decided to choose 32 as the number of hidden units for following experiment.We apply this number to hidden layer configuration of both the Phased-LSTM model and the Phased-LSTM-D model for implementation of the compared five methods.

| Method | Accuracy | AUC | Loss |
|---|---|---|---|
| PLSTM-Forward | 86.40 | 71.85 | 0.438 |
| PLSTM-Linear | 86.44 | 71.86 | 0.402 |
| PLSTM-All | 86.66 | 75.61 | 0.357 |
| PLSTM-Masking | 86.56 | 74.33 | 0.367 |
| PLSTM-D | 86.38 | 73.94 | 0.363 |

*Table 1.* Accuracy, AUC, and value of the loss function of the test dataset.

Performance evaluation of the five methods on the PhysioNet Challenge 2012 dataset with respect to average accuracy, AUC and the loss function value is shown in Table 1. Since we choose the number of hidden units with model training and evaluation to reduce the overfitting, the overfitting problem in the performance evaluation is not obvious to observe. We present the performance on the the test data here. We can observe that among five methods, Phased-LSTM-ALL, Phased-LSTM-Masking and Phased-LSTM-D achieve better performance than the other two methods. Phased-LSTM-ALL achieve the highest accuracy $(86.66\%)$, AUC $(75.61\%)$ and the lowest loss function $(0.357)$ value on the dataset. The next-best methods is Phased-LSTM-Masking with accuracy $(86.56\%)$, AUC $(74.33\%)$ and the loss function $(0.367)$. Phased-LSTM-D has accuracy $(86.38\%)$, AUC $(73.94\%)$ and the loss function $(0.363)$. All methods can achieve high prediction accuracy (around $86\%$) and the difference is less than $0.3\%$ among different methods. Fig 6 shows the change of test accuracy and AUC during training process for Phased-LSTM-D method. The test accuracy increases at first 15 epochs, and then fluctuates around $86.2\%$ after 30 epochs. The test AUC increases fast for the first 10 epochs, and

is flattened after that. It eventually converges at around 74%. We also observed that during the cross-validation of Phased-LSTM-D method, sometimes the AUC performance will still increase after 50 epochs. This suggests that there is room for further improvement of Phased-LSTM-D if we train the model for longer epochs. Figure 7 shows the ROC curves of different methods for testing data. Similar to results shown in Table 1, Phased-LSTM-ALL gives the highest AUC and Phased-LSTM-Masking and Phased-LSTM-D achieve comparable AUC performance. Phased-LSTM-Forward and Phased-LSTM-Linear gives the lowest AUC score.

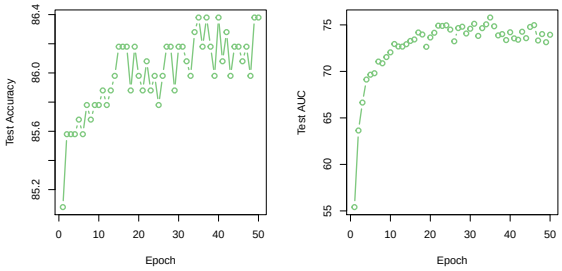



*Figure 6.* Accuracy and AUC during training process for Phased-LSTM-D

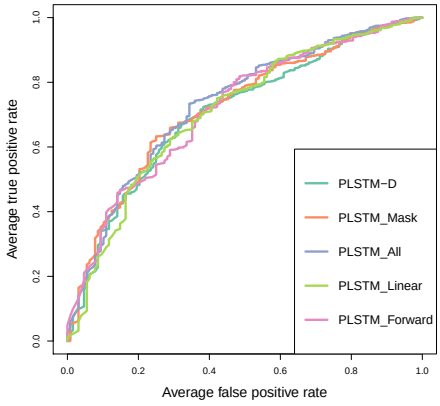



*Figure 7.* ROC curve of different methods

## 5. Conclusion

We learn predictive models based on Phased-LSTM for disease diagnosis and prediction using EHR data. The main challenge is to deal with two types of missing patterns of EHRs: (1) irregularity of sampling intervals and (2) asynchronism of sampling features. Combination of the two missing patterns induced from irregularly sampled subjects and asynchronously sampled features makes it complicate to learn predictive model from EHRs.

In order to deal with the problem, we use two approaches: (1) to impute the missing features first, and then fit Phased-LSTM to deal with the irregularly sampled intervals, and (2) to learn Phased-LSTM-D to simultaneously deal with the irregularity sampling and asynchronism of feature sampling during training. The first type of approach include Phased-LSTM-Forward, Phased-LSTM-Linear, Phased-LSTM-Masking and Phased-LSTM-ALL, which are named based on the imputation method used to replace the missing values. The second approach is the new model *Phased-LSTM-D* that embeds the decay rates $\gamma_t$ suggested by Che et al. (2016) into Phased-LSTM. Phased-LSTM-D takes advantages of both Phased-LSTM and GRU-D. The cell state $c_t$ allows information to pass the LSTM cell so it can capture long range dependencies. The new time gate $k_t$ makes to efficiently learn from the irregularly sampled time-series data generated in continuous time as it does in Phased-LSTM. In order to deal with asynchronously sampled features, it also introduces the decay rates $\gamma_t$ to impute inputs and update hidden states as it does in GRU-D.

We applied our methods to a real EHR dataset (PhysioNet Challenge 2012). Among the five methods, Phased-LSTM-ALL achieves the best performance. Our proposed method Phased-LSTM-D, although not the overall best performance method, can achieve comparable score to the best imputation methods. During our model training, we noticed the problem of over-fitting. Since we only have records of 4000 patients as input, our model is very likely to be over-fitted as the number of parameters increases. We performed a parameter scanning to reduce the number of parameters used and minimize the effect of over-fitting to the model. In ideal situation, more data points should be collected in order to train the model with better performance. In summary, our result suggests that directly imputing missing values is probably not the best way to handle missing values in EHR data. Instead, adding masking and time interval information as additional features can give us better prediction performance. In addition, we showed that Phased-LSTM based algorithms are capable of handling the problem of irregularly and asynchronously sampled EHR data simultaneously.

## References


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

Che, Zhengping, Purushotham, Sanjay, Cho, Kyunghyun, Sontag, David, and Liu, Yan. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

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

Kreindler, David M and Lumsden, Charles J. The effects of the irregular sample and missing data in time series analysis. *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data*, pp. 135.

Lipton, Zachary C, Kale, David C, Elkan, Charles, and Wetzell, Randall. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Lipton, Zachary C, Kale, David C, and Wetzel, Randall. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. *arXiv preprint arXiv:1606.04130*, 2016.

Neil, Daniel, Pfeiffer, Michael, and Liu, Shih-Chii. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, pp. 3882–3890, 2016.

Olah, Christopher. Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

Rehfeld, Kira, Marwan, Norbert, Heitzig, Jobst, and Kurths, Jürgen. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

Rizopoulos, Dimitris and Ghosh, Pulak. A bayesian semiparametric multivariate joint model for multiple longitudinal outcomes and a time-to-event. *Statistics in medicine*, 30(12):1366–1380, 2011.

Schulam, Peter and Saria, Suchi. Integrative analysis using coupled latent variable models for individualizing prognoses. *Journal of Machine Learning Research*, 17(234): 1–35, 2016.

Silva, Ikaro, Moody, George, Scott, Daniel J, Celi, Leo A, and Mark, Roger G. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *Computing in Cardiology (CinC), 2012*, pp. 245–248. IEEE, 2012.

Wang, Hua, Nie, Feiping, Huang, Heng, Yan, Jingwen, Kim, Sungeun, Risacher, Shannon, Saykin, Andrew, and Shen, Li. High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer's disease progression prediction. In *Advances in Neural Information Processing Systems*, pp. 1277–1285, 2012.

White, Ian R, Royston, Patrick, and Wood, Angela M. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.