Information Sciences 579 (2021) 607–622

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Deep dynamic imputation of clinical time series for mortality prediction

Zhenkun Shi [a,b,c,*], Sen Wang [b], Lin Yue [b], Lixin Pang [e], Xianglin Zuo [a,*], Wanli Zuo [a,*], Xue Li [b,d]

[a] College of Computer Science and Technology, Jilin University, Jilin 130012, China
[b] School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane 4072, Australia
[c] Tianjin Institute of Industrial Biotechnology, Chinese Academy of Sciences, Tianjin 300308, China
[d] Dalian Neusoft University of Information, Dalian 116600, China
[e] Institute of Science and Technology of Hebei Agricultural University, Baoding 071001, China

ARTICLE INFO

Article history:
Received 2 November 2020
Received in revised form 3 August 2021
Accepted 5 August 2021
Available online 09 August 2021

Keywords:
Health informatics
Missing value
Imputation
Mortality prediction

ABSTRACT

Missing values in clinical time-series data are pervasive and inevitable; they not only increase the complexity and difficulty of analyzing the data but also lead to biased results. To tackle these two problems, researchers have been exploring recurrent neural network (RNN)-based methods for detecting how well missing values are addressed with the aim of achieving state-of-the-art performance. However, these methods have two practical drawbacks. 1) Handling time-series data with multiple, irregular, abnormal values is difficult. 2) The patterns that may be present in the missing clinical data are not thoroughly considered. Moreover, to the best of our knowledge, none of these methods have been explicitly designed to dynamically optimize the imputation quality for better performance in the realm of clinical time-series analytics. By considering the quality of imputed values, we propose a 2-step integrated imputation-prediction model based on gated recurrent units (GRUs) for medical prediction tasks. In the first step, the missing values are imputed using a sophisticated model based on a replenished GRU with a hidden state decay mechanism (RGRU-D), which is followed by evaluation through two additional layers. In the second step, the optimized imputed values are used to predict the risk of mortality in critical patients. Our model effectively supplies missing values for the *masking*, *time interval*, *bursty*, and *cumulative missing rate* variables within an integrated deep architecture. Extensive experiments on a real-world ICU dataset demonstrate that our model performs better than the compared methods in terms of the imputation quality and prediction accuracy.

1. Introduction

Multivariate time-series data are ubiquitous in clinical studies. However, these data inevitably have missing observations or missing values, which can significantly affect the performance of downstream applications [1]. A missing value is a data value that is not collected or stored for a variable in the current observation, which leads to an empty cell in the dataset. At worst, missing values can significantly influence the conclusions drawn from the data [2]. There are four main reasons for

* Corresponding authors at: College of Computer Science and Technology, Jilin University, Jilin 130012, China (Z. Shi, X. Zuo, W. Zuo).
E-mail address: shizk14@mails.jlu.edu.cn (Z. Shi).

https://doi.org/10.1016/j.ins.2021.08.016
0020-0255/

missing clinical values: equipment failure; clinicians failing to record the data; no intention to collect the data (e.g., the patient was not suffering from a relevant symptom or comorbidity); or an intention to collect the data, but the features returned to normal, failing to realize data charting [3].

Many procedures exist to address missing values [4]. A straightforward solution is to simply ignore the missing records, but this will change the original data structure. To maintain the original structure of the data, a more conventional method is to replace the missing value with the sample means, but the quality of the means may be exceedingly different from that of the application domains [5]. An alternative method for handling missing values is to implement an imputation technique, such as the k-nearest neighbor (KNN) approach, matrix factorization (MF) [6,7], or multivariate imputation by chained equations (MICE) [8]. However, most of these methods require relatively strong assumptions about the missing values [9,10]. Thus, if the corresponding imputation algorithms are not specifically designed based on the assumptions or the data do not meet the assumptions, then the imputation performance will be suboptimal.

Motivated by rapidly rising costs and continued concerns about variations in the content and quality of healthcare, the science of clinical prediction is steadily evolving. Mortality prediction is one of the essential tasks in healthcare, is important for inferring clinical outcomes and has attracted much attention in both academia [11–14] and industry [15]. With the collection of a large number of electronic health records globally, there is an indispensable need to develop effective models for predicting mortality based on these data. However, all of the existing mortality prediction models suffer from the missing value problem. Therefore, developing a mortality prediction model along with an efficient missing value processing mechanism is important.

In this paper, we propose a 2-step integrated imputation-prediction model based on a recurrent neural network (RNN) to fill in the missing values in multiple, correlated, sequential clinical data. The model is called the Model for Imputation and Prediction (M4IP). M4IP has the following characteristics: 1) it iteratively fills in the missing values with optimized values using an additional evaluation step, and 2) it logically incorporates the optimal imputed values into a unified deep framework for clinical prediction tasks. M4IP allows the RNN to accept *masking*, *time interval*, *cumulative missing rate*, and *bursty* data as inputs. In this way, M4IP can simultaneously evaluate both the imputation quality and prediction performance in a unified framework. Objectively, this research makes the following contributions:

1) **A well-considered imputation mechanism for time-series ICU data.** Our model does not impose specific assumptions and fully considers the patterns within and reasons for the missing values, such as *bursty* or *cumulative missing rate*. Hence, our model is more accurate and applicable to ICU applications, where irregularly sampled, missing, or noisy data are common, than previous methods.
2) **A novel integrated framework that directly imputes missing values with a quality evaluation.** Our RNN-based 2-step imputation-prediction model, which is designed for clinical data, integrates data imputation and data classification into the same process rather than merely tuning the weights for smoothness [12].
3) **Dynamic feature-specific constraints for imputation filtering.** Using feature heterogeneity and diversity, we designed a set of learnable, dynamic feature-specific constraints to filter the imputed values, which makes estimating the missing values more accurate.
4) **Comprehensive experiments and discussion.** To validate the effectiveness of M4IP, we evaluated our model in mortality prediction with respect to four different diseases with the Medical Information Mart for Intensive Care (MIMIC-III) dataset [16], which contains 300+ clinical measurements and 200+ medical treatments. The results show that our model outperforms the state-of-the-art models in terms of both the imputation quality and classification accuracy.

2. Related work

2.1. Clinical missing value imputations

Missing value imputation is a hot research topic in clinical practice and has attracted the attention of many researchers [17–20]. A great deal of the related literature is dedicated to imputing missing values in datasets. Techniques range from simply deleting the offending records to statistical computation methods [21,22] to artificial deep learning methods [23]. Yoon et al.[24] summarized three widely used methods for this problem: interpolation, imputation, and matrix completion. Interpolation methods [9,25] attempt to reconstruct the missing data by capturing the synchronous (nontemporal) relationships in other streams of similar data. Conversely, imputation methods [9,25] are used to reconstruct missing values by capturing the synchronous relationships within the given stream. Furthermore, matrix completion methods [6,26–29] divide the temporal data into slices and then make assumptions about a specific model from the data generation process and/or the missing data pattern. Among these methods, the most common is MICE [30]. MICE first initializes missing values arbitrarily and then estimates each missing variable based on a chain equation — an approach that has been shown to perform well and has therefore been included for comparison in our experiments.

Researchers have attempted to impute the missing values using RNNs [1,12,24,31] with encouraging results. The recurrent components are trained with classification or regression components, which has significantly boosted the accuracy. As mentioned in the Introduction, there are two closely related works in this stream, i.e., the gated recurrent units with a decay mechanism (GRU-D) [12] and bidirectional recurrent imputation for time series (BRITS) [1] methods. GRU-D integrates missing data patterns into the model to improve the prediction. We call this a 1-step method, which means that imputation and

prediction work together without any output or evaluation of the imputed results. The advantage of the 1-step method is its ability to considerably improve the prediction accuracy. However, it cannot ensure the quality of the imputed values. BRITS is a more generic 2-step imputation method that uses a bidirectional recurrent imputation method for time-series data that does not require any specific assumptions to hold. BRITS has achieved promising results for air quality, healthcare, and human activity data. Although BRITS uses an imputation mechanism, its performance is not as effective when contending with time-series data and dynamic time windows; In particular, its efficacy is limited when the sequence has multiple irregular peaks. Moreover, these methods were not specifically designed for ICU data, so they may incorrectly weight certain clinical patterns, such as burst distributions. This is a major problem, as ICU features can vary dramatically over a short period of time, and it explicitly demonstrates that the data are not always smooth.

2.2. Mortality prediction

Mortality prediction is important for inferring clinical outcomes [32–34]. In clinical practice, healthcare workers use medical scoring systems as severity assessment tools for patients, most of which use rule-based methods, such as the Acute Physiology, Age, Chronic Health Evaluation (APACHE) score [35], the Sequential Organ Failure Assessment (SOFA) score [36], and the Simplified Acute Physiology Score III (SAPS 3) [37]. However, all these systems implement fixed clinical decision rules that are largely based on clinical statistics [38], but the measurement indicators are limited to a score tend. For instance, the SOFA score is based on only 11 indicators, whereas the MIMIC-III benchmark dataset includes over 4000 individual indicators. When making a diagnosis among thousands of diseases, a set of indicators in the double digits is inadequate.

Fortunately, advances in deep learning are now enabling mortality risk prediction based on a deeper foundation of symptomatology from electronic health records. In fact, several studies that rely on deep learning techniques to forecast in–hospital mortality risk have significantly improved the quality of acute hospital care, e.g., Rajkomar et al., 2018 and Song et al., 2018 [39,40]. However, these methods do not have efficient mechanisms for handling missing data.

Based on the abovementioned concerns, we propose a 2-step model, namely, M4IP, which is designed for imputing ICU data first by incorporating multiple missing data patterns and multiple imputation strategies with an imputation quality control mechanism and then conducting mortality prediction based on the imputed data.

3. Preliminaries

In this section, we introduce some key concepts in the context of our framework, followed by a formal description of our M4IP method. The notations used in this paper are listed in Table 1.

3.1. Key concept definitions

Definition 1 (*Clinical actions*). In the medical domain, to cure patients, a series of clinical operations exist, such as bedside monitoring, laboratory tests, microbiology tests, and medical treatment. In this paper, these operations are called clinical actions. For example, one of the clinical actions is bedside monitoring. This action consists of measuring the heart rate, respiratory rate, temperature, etc. The clinical actions can be represented by $\boldsymbol{A} = (\boldsymbol{a}_1; \boldsymbol{a}_2; \ldots; \boldsymbol{a}_N) \in \mathbb{R}^D$, which is a concatenated vector, where $\boldsymbol{a}_i$ represents a collection of action features and $dim(\boldsymbol{a}_i)$ is the dimension of action $\boldsymbol{a}_i$. $D = dim(\boldsymbol{a}_1) + dim(\boldsymbol{a}_2) + \ldots + dim(\boldsymbol{a}_N)$.

Table 1
Description of the main notations used in this paper.

Notation	Description
$\boldsymbol{A}$	Clinical actions
$\boldsymbol{X}$	Multivariate medical time series
$\boldsymbol{M}$	Input mask, indicates if the value is missing or not
$\boldsymbol{S}$	Input timestamp
$\boldsymbol{\Delta}$	Missing interval
$\boldsymbol{B}$	Burstiness parameter, bursty
$\boldsymbol{CMR}$	Cumulative missing rate
$\boldsymbol{SW}$	Imputation switch
$\mathscr{L}(\hat{x}, x)$	Loss function
tw	Length of the time window
MAX	Maximum $\boldsymbol{X}_j$ value
MIN	Minimum $\boldsymbol{X}_j$ value
PD	Probability distribution of $\boldsymbol{X}_j$ values

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
d_1	1.8	2.5	2.2	-	-	-	1.9	a_1
d_2	188	194	-	-	168	165	157	a_2
d_3	-	-	65	65	70	72	-	a_3
d_4	10	-	15	23	-	-	22	a_3

The time series X

Fig. 1. Example of multivariate time series with missing values. These time-series data consist of $\boldsymbol{a}_1$ to $\boldsymbol{a}_3$, a total of 3 actions and 4 features ($\boldsymbol{a}_3$ has two features, $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$, which are marked in aquamarine environments). These multivariate time-series data have seven observations.

Definition 2 (*Multivariate medical time series*). For patient p, the corresponding multivariate medical time series can be defined as $\boldsymbol{X}^p = \{\boldsymbol{x}_1^p, \boldsymbol{x}_2^p, \cdots, \boldsymbol{x}_{T^p}^p\} \in \mathbb{R}^{D\times T^p}$, where $\boldsymbol{x}_i^p$ is the multivariate observation of patient p in the i-th time window and T^p is the total number of time windows. Fig. 1 shows an example of multivariate time-series data with missing values for a patient.

Definition 3 (*Mask*). As shown in Fig. 1, missing values are ubiquitous in clinical data. As noted in [12], missing values can provide auxiliary information for learning tasks. To flag these missing values, we use a mask to denote the occurrences of missing values in different time windows. A mask can be represented by a matrix: $\boldsymbol{M}^p = \{\boldsymbol{m}_1^p, \boldsymbol{m}_2^p, \ldots, \boldsymbol{m}_{T^p}^p\} \in \mathbb{R}^{D\times T^p}$ for $\boldsymbol{X}^p$. $X_{i,j}^p$ represents the i-th feature in the j-th time window for patient p, while $M_{i,j}^p \in \{0, 1\}$ denotes that the i-th feature is missing in the j-th time window.

$$M_{i,j}^p = \begin{cases} 1, & \text{if } X_{i,j}^p \text{ is observed} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Definition 4 (*Imputation switch vector*). In real-world scenarios, clinical actions may require different imputation strategies. For example, the values of a medical treatment (*e.g.*, action $\boldsymbol{a}_3$ in Fig. 1) can be missing because there was no medicine prescribed to and taken by the patient during the time period. The missing values need to be imputed with 0 s to make sense from a medical point of view. In contrast, the missing values for heart rate records (*e.g.*, action $\boldsymbol{a}_1$ in Fig. 1) require reasonable imputations that effectively reflect the physiological conditions. In this work, the former imputation method is treated as a static imputation method, while the latter is regarded as a dynamic imputation method, which is learnable. To incorporate two different strategies in the imputation procedure, we design an imputation switch vector $\boldsymbol{sw} = (sw_1, sw_2, \ldots, sw_N) \in \mathbb{R}^N$ to choose the imputation method for each action. The entry of the switch vector, sw_i, is defined by:

$$sw_i = \begin{cases} 1, & \text{if action } \boldsymbol{a}_i \text{ requires dynamic imputation} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Definition 5 (*Charting timestamp matrix*). To align the charting time of each action with different time windows, we use a matrix $S^p \in \mathbb{R}^{D\times T^p}$, as shown in Fig. 4, to denote the actual charting time for each feature. For those features that are charted multiple times within one time window, we use the most recent value. In this way, more accurate charting times are preserved. Moreover, the missing interval between two valid records, which is regarded as another important missing data pattern [12], can be more precisely calculated than in previous works [1,12], where only the number of time windows between two valid values is available. Our proposed work can utilize the exact interval time as additional information when considering the decay effect.

Definition 6 (*Missing interval*). In healthcare, the influence of the last-observed input variables diminishes over time when the subsequent values are missing [41]. To capture the relationship between the missing value and the duration of its missingness, in this work, we introduce a vector, $\boldsymbol{\delta}_j^p \in \mathbb{R}^D, j = \{1, 2, \cdots, T^p\}$, to capture the duration of missing values at time step j since the last-observed values (LOs) for each patient. Thus, the entire representation of missing intervals can be depicted by a matrix $\boldsymbol{\Delta}^p = \left\{\boldsymbol{\delta}_1^p, \boldsymbol{\delta}_2^p, \ldots, \boldsymbol{\delta}_{T_p}^p\right\} \in \mathbb{R}^{D\times T^p}$, which helps us record the length of time that the values are missing. To calculate the missing intervals, we assume that the start time and the end time for the j-th time window are tws_j and twe_j, respectively. The length of the time window is $tw = twe_j - tws_j$. The time stamps of the charted features are recorded in a matrix S^p. Thus, the missing interval, $\Delta_{i,j}^p$, of the i-v feature in the j-th time window for the p-th patient is calculated as follows:

$$\Delta_{i,j}^{p} = \begin{cases} 0 & \text{if } M_{i,j}^{p} = 1 \\ tw & \text{if } j = 1 \text{ and } M_{i,1}^{p} = 0 \\ twe_j - S_{i,j-1}^{p} & \text{if } j > 1 \text{ and } M_{i,j}^{p} = 0 \text{ and } M_{i,j-1}^{p} = 1 \\ 1tw + \Delta_{i,j-1}^{p} & \text{if } j > 1 \text{ and } M_{i,j}^{p} = 0 \text{ and } M_{i,j-1}^{p} = 0. \end{cases} \tag{3}$$

Taking as an example feature d_1 from Fig. 5, the values of d_1 are missing in the 4-th, 5-th, and 6-th time windows. According to Eq. 3, the corresponding missing intervals for d_1 are $\Delta_{1,4}^{p}$ ($twe_4 - S_{1,3}^{p} = 35$ mins), $\Delta_{1,5}^{p}$ ($twe_5 - twe_4 + \Delta_{1,4}^{p} = 65$ mins), and $\Delta_{1,6}$ ($twe_6 - twe_5 + \Delta_{1,5}^{p} = 95$ mins), respectively. Time intervals can record the exact missing data duration for all the multivariate time-series data, which can provide a more precise missing data pattern for training.

Definition 7 (*The time-series data burstiness parameter*). The burstiness of temporal data is a popular measure in network analysis and time-series anomaly detection [11], where high values of burstiness indicate the presence of rapidly occurring events in a short period. In medical time-series data, burstiness is an important pattern that characterizes the intensity of feature changes as well as the severity of patient illness. For example, under normal circumstances, a patient's heart rate is close to its mean, while during the critical period, the heart rate can vary to a greater extent. During the whole ICU stay of a patient, the length of the critical period is shorter than that of the normal period. Inspired by related works [11,42], we introduce a burstiness parameter, namely, bursty, to help us capture the irregular clinical patterns in a clinical feature across multiple events. In other words, bursts can capture the temporality of medical features and provide information about the overall dynamics of patients' conditions, which can improve the imputation performance and mortality prediction results. We define the bursty matrix for patient p as $\boldsymbol{B}^{p} = \left\{\boldsymbol{b}_1^{p}, \boldsymbol{b}_2^{p}, \ldots, \boldsymbol{b}_{T^p}^{p}\right\} \in \mathbb{R}^{D \times T^p}$, where $\boldsymbol{b}_j^{p}$ is the bursty vector for the j-th time window. Let $\mu_{i,j}^{p}$ and $\sigma_{i,j}^{p}$ be the mean and standard deviation of $\left\{X_{i,1}^{p}, X_{i,2}^{p}, \ldots, X_{i,j}^{p}\right\}$, respectively. Then, a patient's burstiness for the i-th feature in the j-th time window can be calculated as:

$$B_{i,j}^{p} = \begin{cases} \frac{\sigma_{i,j}^{p} - \mu_{i,j}^{p}}{\sigma_{i,j}^{p} + \mu_{i,j}^{p} + \gamma} & j > 1 \\ -1 & j = 1, \end{cases} \tag{4}$$

where $\gamma = 1.4\text{e} - 45$ to avoid dividing by zero. We set $B_{i,j}^{p} = -1$ if there are no values up to the j-th time window. $B_{i,j}^{p}$ can take a value ranging between -1 and 1. Specifically, $B_{i,j}^{p} = -1$ indicates a periodic sequence, while $B_{i,j}^{p} = 0$ indicates a Poisson distribution in the intervening sequence. When $B_{i,j}^{p}$ reaches 1, the sequence becomes more bursty [42]. For example, the burstiness of feature $\boldsymbol{d}_3$ (in Fig. 6) in the 3-rd and 4-th time windows is $B_{3,3}^{p} = B_{3,4}^{p} = -1$, which indicates that there are no value changes to $\boldsymbol{d}_3$ in the 3-rd and 4-th time windows. Afterwards, $\boldsymbol{d}_3$ increases in the 5-th and 6-th time windows ($-1 < B_{3,5}^{p} < B_{3,6}^{p}$), which indicates that the variation in the 6-th time window is greater than that in the 5-th time window. As shown in Fig. 6, we use the cumulative burstiness and ignore the missing values when calculating the burstiness. That is, we use only the values up to and including the current observation to calculate $\sigma_{i,j}^{p}$ and $\mu_{i,j}^{p}$; if $X_{i,j}^{p}$ is missing, we replace it with the latest calculated value according to Eq. 4.

Definition 8 (*Cumulative missing rate (CMR)*). The last missing data pattern in medical time-series data considered in this work is the CMR for each feature. The missing rate of the features in the dataset that are frequently used is particularly useful because the imputation difficulty is different between low and high missing rates. A lower missing rate can result in more sample observations, while higher rates limit the number of samples. Therefore, we propose calculating the missing rates for features in a cumulative manner and embedding them into M4IP as additional information, aiming to improve the imputation and mortality prediction performance. Given a multivariate time series $\boldsymbol{X}^{p}$, for the i-th feature in the j-th time window, the total number of variables we can observe up to the j-th ($j \geqslant 1$) time window is α, and the corresponding CMR is defined as follows:

$$CMR_{i,j}^{p} = \begin{cases} \frac{\alpha}{j}, & \text{if } M_{i,j}^{p} = 0 \\ 0, & \text{if } M_{i,j}^{p} = 1. \end{cases} \tag{5}$$

4. Problem description

In this work, we aim to handle the missing value problem in clinical time-series data with quality assurance. Many metrics can be used to address clinically missing values (*e.g.*, MICE [8] and MF [43]). However, they often require strong assumptions, such as missing completely at random (MCAR) [2], smoothing [8], and nonnegative values [43]. As described in the Related Work section, the inherent limitations of these works greatly restricted their usage in clinical scenarios. Therefore, in our work, the proposed M4IP has two steps: in the first step, the missing values are imputed using the abovementioned

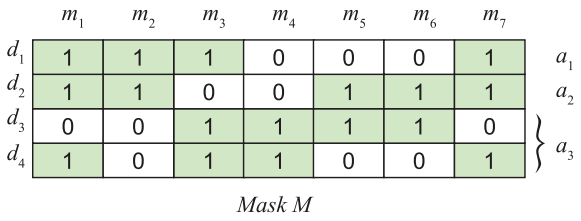

Fig. 2. Example of a missing value mask. A missing value is marked as 0 in $\boldsymbol{M}$; in contrast, if the value exists, then it is marked as 1 in $\boldsymbol{M}$.

patterns, and in the second step, the time-series classification problem is solved. The first step aims to impute the missing values in $\boldsymbol{X}^p$ as accurately as possible; the problem can be formulated as follows:

$$min\left(\|\hat{\boldsymbol{X}}^p - \boldsymbol{X}^p\|_2\right), \tag{6}$$

where $\hat{\boldsymbol{X}}^p$ is the imputed matrix and $\boldsymbol{X}^p$ is the ground truth matrix.

The second step is a traditional classification problem. In this paper, we used the case of mortality predictions for ICU stays of 30 days or less. To integrate the classifications into the imputed network, a classification label $l^p \in \{0, 1\}$ is introduced for $\boldsymbol{X}^p$, where $l^p = 1$ indicates that the patient dies within 30 days after admission to the ICU. The mortality prediction goal is to predict the label l^p by using M4IP based on $\boldsymbol{X}^p$.

Input The basic input is the multivariate time-series data $\boldsymbol{X}^p$ (Fig. 1), with missing values, imputation switch $\boldsymbol{SW}$ (Fig. 3) and timestamp data $\boldsymbol{S}^p$ (Fig. 4). To fully describe the clinical missing data patterns, we add four calculated matrices $\boldsymbol{M}^p$ (Fig. 2), $\boldsymbol{\Delta}^p$ (Fig. 5), $\boldsymbol{B}^p$ (Fig. 6) and $\boldsymbol{CMR}^p$ (Fig. 7) as replenishment inputs:

$$\boldsymbol{I} = \left\{\boldsymbol{X}^p; \boldsymbol{SW}; \boldsymbol{S}^p; \boldsymbol{M}^p; \boldsymbol{\Delta}^p; \boldsymbol{B}^p; \boldsymbol{CMR}^p\right\} \in \mathbb{R}^{7D \times T^p}. \tag{7}$$

As illustrated in Figs. 3–8, the basic inputs are input time series $\boldsymbol{X}$, imputation switch $\boldsymbol{SW}$ and input timestamps $\boldsymbol{S}$. The replenishment inputs are the mask $\boldsymbol{M}$, time interval $\boldsymbol{\Delta}$, bursty $\boldsymbol{B}$ and cumulative missing rate $\boldsymbol{CMR}$ parameters, and the replenishment inputs can be calculated by using Algorithm 1.

Algorithm 1: Calculation of the replenishment inputs

Input : $\boldsymbol{X}^p$, $\boldsymbol{S}^p$, $\boldsymbol{SW}$.

Output: Final input concatenated matrix, $\boldsymbol{G}^p$.

begin

  Construct the switch matrix $\boldsymbol{S\hat{W}} \in \mathbb{R}^{D \times T^p}$, where all the feature values in the entire time window are initialized.

  Construct the mask matrix $\boldsymbol{M}^p$ according to Equation (1).

  Construct $\boldsymbol{\Delta}$ according to Equation (3).

  Construct $\boldsymbol{CMR}^p$ according to Equation (5).

  Construct the bursty matrix $\boldsymbol{B}^p$ according to Equation (4).

  Add $\boldsymbol{S\hat{W}}$ to the mask matrix $\boldsymbol{M}^p$:

  $\boldsymbol{M}^p \leftarrow \boldsymbol{M}^p + \sim \boldsymbol{S\hat{W}}$ ($\sim$ is the binary negation operation).

  Add $\boldsymbol{S\hat{W}}$ to $\boldsymbol{\Delta}^p$: $\boldsymbol{\Delta}^p \leftarrow \boldsymbol{\Delta}^p \cdot \boldsymbol{S\hat{W}}$.

  Add $\boldsymbol{S\hat{W}}$ to the bursty matrix $\boldsymbol{B}^p$: $\boldsymbol{B}^p \leftarrow \boldsymbol{B}^p \cdot \boldsymbol{S\hat{W}}$.

  Add $\boldsymbol{S\hat{W}}$ to $\boldsymbol{CMR}^p$: $\boldsymbol{CMR}^p \leftarrow \boldsymbol{CMR}^p \cdot \boldsymbol{S\hat{W}}$.

  Add $\boldsymbol{S\hat{W}}$ to the timestamp matrix $\boldsymbol{S}^p$: $\boldsymbol{S}^p \leftarrow \boldsymbol{S}^p \cdot \boldsymbol{S\hat{W}}$.

  Concatenate the abovementioned matrices

  $\boldsymbol{G}^p = \{\boldsymbol{X}^p; \boldsymbol{S}^p; \boldsymbol{M}^p; \boldsymbol{\Delta}^p; \boldsymbol{B}^p; \boldsymbol{CMR}^p\}$.

end

return $\boldsymbol{G}^p$.

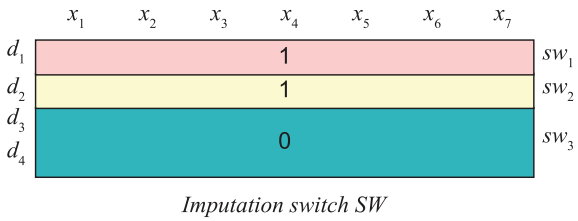

Fig. 3. Example of an imputation switch vector. Assuming action $\boldsymbol{a}_1$ is heart rate monitoring and action $\boldsymbol{a}_3$ contains two input medical features, the corresponding entries in the switch vector that control the imputation strategies are $sw_1 = 1$ and $sw_3 = 0$, respectively. In other words, $sw_1 = 1$ indicates that missing values in action $\boldsymbol{a}_1$ require dynamic imputation, while $sw_3 = 0$ means that missing values in action $\boldsymbol{a}_3$ require only static imputed values (i.e., 0 s).

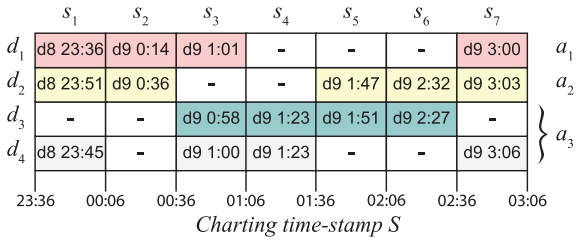

Fig. 4. Example of a charting timestamp matrix $\boldsymbol{S}^p$. $S^p_{i,j}$ records the actual timestamp of the i-th feature in the j-th time window for patient p. For example, $S^p_{2,1}$ indicates that the observation occurs at 23:51 on day 8 after admission, while $S^p_{2,2}$ means that the next value is recorded at 00:36 on day 9. Note that the length of the time window (tw) is 30 min.

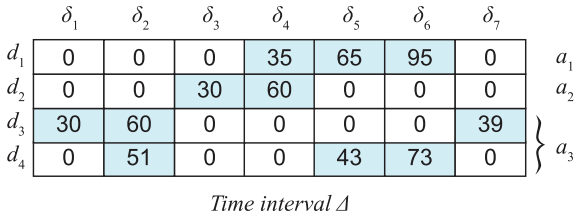

Fig. 5. Example of a missing interval.

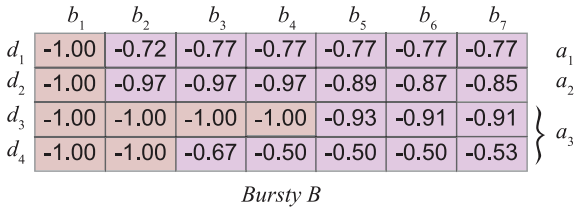

Fig. 6. Example of bursty.

Fig. 7. Example of the CME.

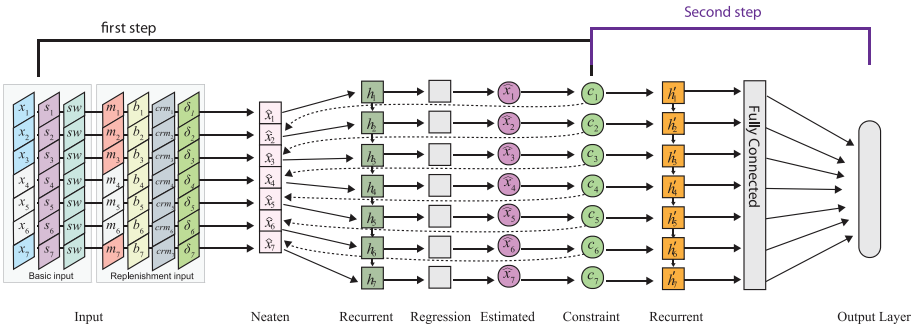

Fig. 8. Proposed 2-step integrated imputation-prediction model (M4IP).

Output There are two kinds of output, the imputation output and the mortality prediction output. For the first $(T-1)$-th step, we give the imputation output, and we output the mortality prediction results only in the final step T. The first output is used to impute the missing values, and we can add a constraint to these imputed values. This output can also be verified by domain experts (*e.g.*, clinicians and nurses). Therefore, the imputed data can have a higher confidence in terms of quality assurance. The second output is based on a specific prediction task; in this paper, we make mortality predictions, which is a binary classification task.

5. Proposed approach

The architecture is developed based on a 2-step strategy. First, the missing values are learned directly in a dynamic (recurrent) way based on the current values. Then, the imputed data were used to make the mortality predictions. There are two distinct advantages of this approach. 1) The missing values are imputed according to the recurrent dynamics and can be shown to the clinicians to evaluate the patient's status more precisely. 2) The prediction performance can be significantly boosted by basing the predictions on imputed data. As shown in Fig. 8, the integrated network consists of seven parts: an input layer, a neaten layer, a recurrent layer, a regression layer, an estimated layer, a constraint layer, and an output layer.

5.1. The input layer

In the input layer, we expand N actions into D features, which include three original inputs, namely, 1) the time-series data $\boldsymbol{X}^p$, 2) adjoint data charting timestamp matrix $\boldsymbol{S}^p$, and 3) imputation switch matrix $\boldsymbol{SW}$, and four calculated inputs as replenishment input matrices, namely, 1) the mask data $\boldsymbol{M}^p$, 2) bursty data $\boldsymbol{B}^p$, 3) cumulative missing rate $\boldsymbol{CMR}^p$, and 4) time interval $\boldsymbol{\Delta}^p$. The original inputs can be directly obtained from the clinical time-series data. The replenishment inputs can be calculated by using Algorithm 1.

5.2. The neaten layer

As time-series data $\boldsymbol{X}^p$ contain missing values, they cannot be directly transferred to the RNN. The neaten layer is the first imputation step and is designed to contend with missing values. To provide an initial value to the missing data, we have considered three different aspects: **1) Different actions require different imputation strategies.** Bedside sensors and medical treatments were taken as examples. At the i-th observation, the temperature information for patient p is missing, so we should replace the missing value with an imputed value. However, this imputation strategy should not apply to missing values in medical treatment records. **2) For the same action, different features require different imputation strategies.** For example, in the action of bedside monitoring, the temperature tends to the sample mean, while the pressure of the pulmonary artery tends to the population mean [44,45]. The sample mean of the temperature is the average value of one patient's temperatures, and the population mean is the average value of all patients' temperatures. **3) For a particular feature, the imputation strategy should vary over time for a particular feature.** During the critical period, the feature value is likely to reach its extreme. In contrast, the feature is more likely to hover around its mean in a normal situation. For example, the average normal body temperature is generally accepted to be between 36.5 and 37 °C, but for a patient with a fever, the body temperature is more likely to be over 38 °C. Thus, during a patient's admission, the imputation strategy should dynamically consider illness severity over time to reflect the true conditions.

With Eq. (1), we can use the following method to give an initial value to the missing data:

$$X_{i,j}^p \longleftarrow M_{i,j}^p X_{i,j}^p + \left(1 - M_{i,j}^p\right)\widehat{X}_{i,j}^p, \tag{8}$$

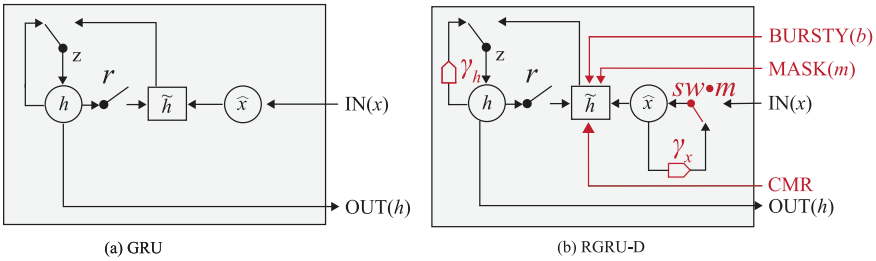

Fig. 9. Illustrations of the GRU and RGRU-D.

where $\widehat{X}_{ij}^{p}$ is the imputed value obtained by our pretrained method $P(x)$. $P(x)$ uses one of eight basic imputation methods: the sample mean (SM), median (ME), population mean (PM), KNN, LO or MICE:

$$P(x) = min(SM(x), ME(x), PM(x), \quad KNN(x), LO(x), MICE(x)). \tag{9}$$

According to our imputation strategies, basic imputation is an unsupervised method that is designed for each missing element X_{ij}^{p}. By adding an imputation switch ***SW*** that can be flipped, the final state of the initial imputation can be calculated as follows:

$$X_{ij}^{p} \longleftarrow M_{ij}^{p} X_{ij}^{p} + SW_{ij}\left(1 - M_{ij}^{p}\right)\widehat{X}_{ij}^{p}. \tag{10}$$

According to the bidirectional correlated recurrent imputation mechanism in the backpropagation process, we can iteratively impute the missing values in the sequence using a loss function $\mathscr{L}\left(\widehat{X}_{ij}^{p}, X_{ij}^{p}\right)$, which is based on the historical data and measurements of the neighbors in each time window j.

5.3. The recurrent and regression layers

These two layers, together with the estimated layer, are the main components of the imputation process. The recurrent component is achieved by an RNN, and the regression layer is achieved by a fully connected network. We adopted a replenished GRU with a hidden state decay mechanism, named RGRU-D, for the recurrent layer. The structures of the GRU and RGRU-D are shown in Fig. 9.

The typical structure of the GRU is shown in Fig. 9(a). For each j-th hidden unit, the GRU has a reset gate r_j and an update gate z_j to control the hidden state h_j at each observation j. Formally, the initial hidden state h_j^0 is initialized as a vector of zeros, and then, the model is updated by:

$$\boldsymbol{r}_j = \sigma\left(\boldsymbol{W}_r \boldsymbol{x}_j + \boldsymbol{U}_r \boldsymbol{h}_{j-1} + \boldsymbol{b}_r\right) \tag{11}$$

$$\boldsymbol{z}_j = \sigma\left(\boldsymbol{W}_z \boldsymbol{x}_j + \boldsymbol{U}_z \boldsymbol{h}_{j-1} + \boldsymbol{b}_z\right) \tag{12}$$

$$\tilde{\boldsymbol{h}}_j = tanh\left(\boldsymbol{W}\boldsymbol{x}_j + \boldsymbol{U}\left(\boldsymbol{r}_j \odot \boldsymbol{h}_{j-1}\right) + \boldsymbol{b}\right) \tag{13}$$

$$\boldsymbol{h}_j = \left(1 - \boldsymbol{z}_j\right) \odot \boldsymbol{h}_{j-1} + \boldsymbol{z}_j \odot \tilde{\boldsymbol{h}}_j, \tag{14}$$

where matrices $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}, \boldsymbol{U}_z, \boldsymbol{U}_r$, and $\boldsymbol{U}$ and vectors $\boldsymbol{b}_z, \boldsymbol{b}_r$, and $\boldsymbol{b}$ are the model parameters, $\sigma(\cdot)$ is an elementwise sigmoid function, and $\odot$ is elementwise multiplication.

Two important properties have been observed in the healthcare domain. First, the values of a missing variable tend to be the default value if the last observation was recorded quite a long time ago [46]. Second, the influence of the last-observed variables diminishes over time when the following values are missing [41]. As shown in Fig. 9(b), RGRU-D adds two trainable decay components (γ_x and γ_h) to the input and hidden layers. A decay rate vector $\boldsymbol{\gamma}$ is defined as follows:

$$\boldsymbol{\gamma}_j = exp\left\{-max\left(0, \boldsymbol{W}_\gamma \boldsymbol{\delta}_j + \boldsymbol{b}_\gamma\right)\right\}, \tag{15}$$

where $\boldsymbol{W}_\gamma$ and $\boldsymbol{b}_\gamma$ are trained on the data along with the other parameters. First, $\boldsymbol{\gamma}_j$ decays the input over time toward its empirical mean:

$$\widetilde{X}_{ij} \longleftarrow M_{ij} X_{ij} + SW_i\left(1 - M_{ij}\right)\gamma_{ij} X_{i'j} + SW_i\left(1 - M_{ij}\right)\left(1 - \gamma_{ij}\right)\overline{X}_{ij}, \tag{16}$$

where $X_{i'j}$ is the last value observed for the i-th feature, SW_i is the imputation switch for the i-th feature, and $\overline{X}_{ij}$ is its sample mean. Second, RGRU-D adds a temporal decay factor $\boldsymbol{\gamma}_j$ to the hidden state, and the decay function is:

$$\tilde{\boldsymbol{h}}_{j-1} \longleftarrow \boldsymbol{\gamma}_j \odot \boldsymbol{h}_{j-1}. \tag{17}$$

Next, the RGRU-D adds three replenishment sources directly to the hidden state (marked in red) to capture the clinical patterns in multivariate time-series data individually: a mask matrix $\boldsymbol{M}$, a bursty matrix $\boldsymbol{B}$, and a $\boldsymbol{CMR}$ matrix denoted by $\boldsymbol{C}$. By introducing these replenishment sources, $\tilde{h}_i$ can be redefined as:

$$\tilde{\boldsymbol{h}}_j = tanh\left(\boldsymbol{W}\tilde{\boldsymbol{x}}_j + \boldsymbol{U}\left(\boldsymbol{\gamma}_j \tilde{\boldsymbol{h}}_{j-1}\right) + \boldsymbol{V}_m \boldsymbol{m}_j + \boldsymbol{V}_b \boldsymbol{b}_j + \boldsymbol{V}_\delta \boldsymbol{\delta}_j + \boldsymbol{V}_c \boldsymbol{c}_j + \boldsymbol{b}\right), \tag{18}$$

where $\tilde{\boldsymbol{x}}_j$ can be calculated with Eq. (16).

5.4. The estimated layer

This layer is based on the context of an ICU and the fact that every part of the human body is closely related; therefore, there is a great deal of interplay between the outcomes of many measures. In this work, we produce two kinds of estimations in this layer. The first is feature-related (FR) estimation, in the j-th time window, for two features $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$ ($a \neq b$). Similar to FR estimation, the second type of estimation is temporal correlation within one measurable variable. For example, the heart rate of a patient should be temporally correlated over a period of time. We call this estimation type history-related (HR) estimation. This type can be formulated as follows: for the j-th and k-th time windows, where $j \neq k, \boldsymbol{X}_{\cdot,j}$ and $\boldsymbol{X}_{\cdot,k}$ should be correlated.

For FR estimation, we first obtain the complement observation $\tilde{\boldsymbol{x}}_j$ by M4IP, and then, we define our FR estimate as $\boldsymbol{q}_j$:

$$\boldsymbol{q}_j = \boldsymbol{W}_q \tilde{\boldsymbol{x}}_j + \boldsymbol{b}_q, \tag{19}$$

where $\boldsymbol{W}_q$ and $\boldsymbol{b}_a$ are corresponding parameters, which are obtained from their regression layers. We restrict the diagonal of the parameter matrix $\boldsymbol{W}_q$ to zeros. Thus, the i-th element in $\boldsymbol{q}_j$ is exactly the estimation of $X_{i,j}$ based on the other features.

For HR estimation, we produce an estimation sequence $\boldsymbol{X}^+ = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \cdots, \boldsymbol{x}_T^+\} \in \mathbb{R}^{D \times T}$ in the forward direction, accompanied by a loss sequence $\{\ell_1^+, \ell_2^+, \cdots, \ell_T^+\}$. Similarly, in the backward direction, we obtain another estimation sequence $\boldsymbol{X}^- = \{\boldsymbol{x}_1^-, \boldsymbol{x}_2^-, \cdots, \boldsymbol{x}_T^-\} \in \mathbb{R}^{D \times T}$ with another loss sequence $\{\ell_1^-, \ell_2^-, \cdots, \ell_T^-\}$. We enforce the prediction in each step to be consistent in both directions by using the distance between the forward direction and the backward direction:

$$\ell_j^{di} = \|\boldsymbol{x}_j^- - \boldsymbol{x}_j^+\|_2. \tag{20}$$

The final estimation loss term for HR estimation is obtained by accumulating the forward loss ℓ^+, the backward loss ℓ^-, and the distance loss ℓ^{di} over T time windows. Thus, the RGRU-D aims to minimize the accumulated loss for each data point: $\sum_{j=1}^{T} \ell_j$, where $\ell_j = \|\boldsymbol{x}_j^+ - \boldsymbol{x}_j\|_2 + \|\boldsymbol{x}_j^- - \boldsymbol{x}_j\|_2 + \|\ell_j^{di} - \boldsymbol{x}_j\|_2 + \|\boldsymbol{q}_j - \boldsymbol{x}_j\|_2$.

5.5. The constraint operation

Constraint operation allows the use of hand-engineered rules to constrain the imputed values. For example, common sense dictates that the heart rate must be between 0 and 300. Here, a prelearned feature probability distribution is used to constrain the imputed values. In this work, we use a prelearned sampling probability distribution, which is also based on clinical action-based facts, to constrain the imputed data, as the statistical patterns derived from the population data are very helpful when constraining the imputed values [47]. Therefore, population-based feature distribution patterns are efficient in constraining the imputed features. Inspired by Joseph et al., we use the three most common distributions: binomial, normal, and Poisson. In addition, we also use the maximum and minimum values to further constrain the imputation results.

After the constraint operation, the initial imputed values $\widehat{X}_{i,j}^p$ can be updated with the output results $\widetilde{X}_{i,j}^p$.

$$\widehat{X}_{i,j}^p \longleftarrow \widetilde{X}_{i,j}^p, \tag{21}$$

where $\widetilde{X}_{i,j}^p$ is the imputed value after the constraint operation.

5.6. Mortality prediction

To predict the mortality of ICU patients, we adopt the cross-entropy loss with L_2 regularization as the loss function for binary classification. To solve the data imbalance problem, we add a weighting parameter to penalize mistakes in the minority class. More specifically, errors relative to class k ($k \in \{1, \ldots, K\}$) are weighted by the term $c_k = 1 - N_k/N$, where N_k is the number of training samples of class k and N is the size of the training set. In this way, classification errors in the class with fewer elements contribute more than errors in the other class. The resulting loss function is:

$$\mathscr{L} = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k} c_k\left[y_i^k log(\hat{y}_i^k) + (1-y_i^k) log(1-\hat{y}_i^k)\right] + \lambda\|W\|_2, \tag{22}$$

where y_i^k and $\hat{y}_i^k$ are the ground truth and prediction, respectively, for the k-th class. Notably, there are only two classes in mortality prediction, and the class distribution is usually imbalanced. In other words, there are much fewer positive observations (i.e., death records) than negative observations. $\|W\|_2$ is the L_2 norm of all the network weights, and λ weights the regularization strength.

6. Experiment

6.1. Dataset and experiment settings

We conducted our experiments on the MIMIC-III real-world ICU dataset containing data on four different diseases. MIMIC is a publicly available dataset containing deidentified health data associated with more than 40,000 critical care patients and approximately 60,000 ICU stays.

We first categorized the ICU stays based on codes from the 9th version of the International Classification of Diseases (ICD9) [48]. We then chose the four most commonly diagnosed diseases: hypertension (ICD: 4019), coronary atherosclerosis (ICD: 41401), renal failure (ICD: 5849), and diabetes mellitus (ICD: 25000). For each disease, we counted the clinical measurement features, and the medical treatment features were treated in descending order. We selected the top 300 $f_{measure}$ and top 200 f_{treat} for inclusion in the data. Next, we randomly hid 5%, 10%, 15%, 20%, 30% of the observed values as the ground truth. We trained our model with an Adam optimizer at a learning rate of 0.001 and a batch size of 400. We also implemented an early stopping strategy given a validation error. We used the pretrained model from Step 1 for the classification step, followed by 10-fold cross-validation to further optimize both the imputation and classification losses simultaneously.

6.2. Baseline methods

To test the imputation step, we compared M4IP with both RNN-based methods and non-RNN-based methods. 1) Sample mean: replace the missing values with the corresponding sample mean; 2) KNN: imputes missing values using the KNN algorithm [49]; 3) MF: factorizes the data matrix into two low-rank matrices and fills in the missing values using matrix completion [26]; 4) LO: replaces a missing value with the last observation (and, for simplicity, 0 if no previous observation exists, which is a common situation in ICU data); 5) MICE: we use a method named Pandas-MICE, which is the Python version of MICE [8]; 6) GRU-D: we use a reimplementation of Zhenpin Che et al.'s work [12], which employs an RNN for multivariate time series with missing values; 7) BRITS: we use a standard implementation of BRITS [1].

To evaluate the predictions, we replaced our imputation method with the baseline methods and then integrated them into the second step. This approach meant that we could compare all the methods under the same experimental settings.

GRU-D, BRITS, and M4IP were implemented using PyTorch with two GTX 2080 Ti GPUs.

6.3. Evaluation metrics

We used two metrics to assess the imputation results: the mean absolute error (MAE) and mean relative error (MRE), defined as follows:

$$MAE = \frac{\sum_i|\tilde{y}_i - y_i|}{N} \tag{23}$$

$$MRE = \frac{\sum_i|\tilde{y}_i - y_i|}{\sum_i|y_i|}, \tag{24}$$

where $\tilde{y}_i$ is the imputed value, y_i is the ground truth of the i-th item, and N is the total number of imputed items. We used three metrics to evaluate the performance of the mortality prediction: accuracy (ACC), precision, and area under the precision-recall curve (AUPRC).

6.4. Experimental results and discussion

6.4.1. Imputation results

Tables 2 and 3 show the imputation performance. From these tables, we can see that the same methods perform differently on different tasks, e.g., tasks 41401 and 5849, where the MAE has a more than seven-point difference. This finding indicates that these methods greatly depend on the amount and quality of the input data. The mean imputation method had lower average values but higher variance among all the patients. The lower mean values indicate that the majority of the features of the patient are regular most of the time. A higher standard deviation indicates that the method is not suitable for uneven values, which frequently occur in ICU data, and this also indicates that adding a burst pattern is useful. The fea-

Table 2
Performance comparison for the imputation tasks – mean absolute error (MAE).

Category	Method	4019	41401	5849	25000
Non-RNN	LO	16.08 ± 74.96	22.61 ± 116.82	15.06 ± 27.42	16.45 ± 18.40
	Mean	8.62 ± 145.80	8.87 ± 123.54	9.76 ± 311.62	9.90 ± 77.00
	KNN	23.03 ± 32.87	25.72 ± 19.25	17.24 ± 60.24	14.29 ± 6.83
	MICE	19.76 ± 61.97	23.61 ± 21.13	14.96 ± 13.36	13.07 ± 6.07
	MF	18.75 ± 27.87	23.38 ± 7.23	14.72 ± 27.69	13.00 ± 7.21
RNN	BRITS	17.09 ± 6.21	21.62 ± 9.40	11.36 ± 8.36	12.05 ± 18.60
	GRU-D	16.54 ± 7.32	20.55 ± 8.74	9.62 ± 7.29	11.05 ± 11.85
Ours	M4IP	8.73 ± 6.68	8.73 ± 6.68	6.13 ± 4.32	8.29 ± 5.88

Table 3
Performance comparison for the imputation tasks – mean relative error (MRE).

Category	Method	4019	41401	5849	25000
Non-RNN	LO	0.35 ± 0.44	0.31 ± 0.70	0.42 ± 0.21	0.29 ± 0.45
	Mean	0.18 ± 0.69	0.18 ± 0.63	0.19 ± 0.69	0.19 ± 0.69
	KNN	0.39 ± 0.34	0.38 ± 0.43	0.59 ± 0.42	0.29 ± 0.37
	MICE	0.33 ± 0.40	0.36 ± 0.36	0.40 ± 0.24	0.37 ± 0.24
	MF	0.33 ± 0.30	0.36 ± 0.20	0.40 ± 0.18	0.37 ± 0.34
RNN	BRITS	0.33 ± 0.16	0.38 ± 0.19	0.35 ± 0.16	0.32 ± 0.37
	GRU-D	0.32 ± 0.21	0.37 ± 0.17	0.31 ± 0.12	0.29 ± 0.24
Ours	M4IP	0.16 ± 0.15	0.16 ± 0.15	0.12 ± 0.08	0.17 ± 0.14

ture matrix method performed consistently with the low-rank and sparse hypotheses due to the high rate of missing values, while the MF method showed favorable results. We used $k = 2, 3, 4, 5, 6, 8, 10$ for the KNN method and found that the majority of its best performances occurred at $k = 3$ or $k = 4$, which indicates that clinical features are highly correlated with the sequence. From Table 2, we can see that BRITS performed the worst among the three RNN-based methods. This is because BRITS performs poorly when contending with irregular time sequences. GRU-D was designed for health records and adopted a bidirectional RNN, so it performed second best in terms of MAE and MRE. We attempted to impute all the tasks together, but this resulted in the worst performance, indicating that a refined imputation is necessary, such as categorization by disease.

For the imputation evaluation, M4IP achieved the best performance measured by the MAE compared to all the imputation methods. From Table 2, regarding task 4019, the averaged imputation performance of our proposed method is close to that of the mean imputation strategy (M4IP: 8.7338, mean: 8.6214), but the standard deviation of the proposed method is much lower than that of the mean strategy (6.6767 vs 145.7971). Compared to the second-best RNN-based method, GRU-D, our imputation consistently outperforms the other strategies in all tasks. Additionally, we observed a similar performance improvement in Table 3, where the imputation quality is measured by the MRE. Notably, our method can achieve the best performance in terms of the MRE in all the different prediction tasks.

6.4.2. Mortality prediction results

As shown in Table 4, our method, M4IP, has achieved the best performance in terms of **accuracy**, **precision**, and **F1** scores for four different prediction tasks. This indicates that M4IP is effective when addressing the mortality prediction problem.

Table 4
Performance Comparison for Mortality Prediction using M4IP.

Task	Metric	LO	Mean	KNN	MICE	MF	GRU-D	BRITS	M4IP
4019	ACC	0.8031	0.7973	0.7892	0.8112	0.8201	0.8669	0.8352	0.9402
	AUROC	0.7215	0.7282	0.7043	0.7390	0.7389	0.7702	0.7509	0.8746
	AUPRC	0.8557	0.8590	0.8331	0.8609	0.8621	0.9254	0.9184	0.9763
41401	ACC	0.8857	0.8944	0.8642	0.8867	0.8871	0.9402	0.9000	0.9421
	AUROC	0.7387	0.7831	0.7701	0.7961	0.7923	0.7991	0.7749	0.8051
	AUPRC	0.8881	0.9599	0.8923	0.9320	0.9358	0.9670	0.9591	0.9764
5849	ACC	0.8114	0.7602	0.8227	0.8205	0.8223	0.8894	0.8433	0.9378
	AUROC	0.8224	0.8152	0.8014	0.8200	0.8192	0.8690	0.8200	0.8761
	AUPRC	0.8766	0.8925	0.9079	0.9192	0.9212	0.9438	0.9352	0.9741
25000	ACC	0.8601	0.7635	0.8717	0.8777	0.8770	0.8976	0.8600	0.9227
	AUROC	0.8000	0.8107	0.7924	0.8067	0.8092	0.8100	0.7913	0.8204
	AUPRC	0.9242	0.9463	0.9301	0.9502	0.9414	0.9633	0.9510	0.9646

The mortality rate is very different among all of these tasks, which means that the class distribution of the dataset is imbalanced. GRU-D, BRITS, and M4IP achieved better performance than the other methods in terms of the recall measure, demonstrating that RNN-based methods are better at finding positive samples.

Similarly, M4IP also achieved the best AUROC performance in all the tasks, which shows that the proposed method is superior to all the compared methods.

From Table 2 and Table 4, we can obtain two observations: 1). more complicated imputation strategies normally yield better prediction results; 2). RNN-based methods often outperform non-RNN-based methods in prediction tasks. For example, GRU-D, BRITS, and M4IP have been demonstrated to be better imputation strategies (low error in Table 2 compared to LO, mean, MICE, and MF). Meanwhile, their performance is usually better than the performance of non-RNN-based methods. In addition, unlike previous works [12,1], we did not filter any patients in our experiments, and the length of the sequence was dynamic. Thus, M4IP is more flexible and applicable to medical data analysis, where imputation quality control is indispensable.

6.5. Performance comparison

To evaluate the imputation performance between our method and the existing baselines more comprehensively, we conducted a statistical analysis for the imputation results. We first calculated the absolute error (AE) for each imputed value with its corresponding ground truth, and then we performed a statistical analysis for all these values. The results are shown in Table 5. From the results, we can observe the following: 1) Our proposed method achieves the best performance on all datasets. This is reflected by the 25%, 50%, 75%, and maximum AE values, which are much smaller than the corresponding imputed values from the baselines. 2) The imputation quality of the RNN-based methods is better than that of the location-based methods (LO, Mean, KNN, MICE), which is reflected in the maximum absolute value. This outcome occurs

Table 5
Imputation Quality Comparison on Various Imputation Methods.

Task	Method	@25%[1]	@50%[1]	@75%[1]	Max[1]	PCC[2]	P[3]	Improvement[4]
4019	LO	24.47	51.67	88.14	2500	0.7	6.24E−64	84.19%
	Mean	4.68	9.95	16.78	598.77	0.71	7.14E−01	1.26%
	KNN	13.14	28.04	47.05	162.92	0.62	4.63E−181	163.80%
	MICE	20.87	44.01	75.2	252.23	0.64	7.08E−196	126.35%
	MF	11.08	23.55	39.45	117.56	0.67	8.71E−15	114.78%
	BRITS	12.9	17.07	21.24	42.62	0.77	8.51E−01	95.76%
	GRU-D	11.62	16.51	21.45	48.02	0.82	6.48E−16	89.46%
	M4IP	4.62	8.8	13.23	38.46	0.84	−−	−−
41401	LO	37.82	80.1	137.07	2500	0.7	2.20E−88	158.99%
	Mean	3.99	8.37	14.22	476.45	0.72	4.45E−01	1.60%
	KNN	13.86	25.93	38.71	104.99	0.61	9.60E−03	194.62%
	MICE	12.27	24.02	38.03	108.78	0.64	6.48E−16	170.45%
	MF	18.48	23.38	28.3	51.49	0.67	9.97E−66	167.81%
	BRITS	15.29	21.61	27.96	59.47	0.8	5.71E−02	147.65%
	GRU-D	14.6	20.53	26.43	58.72	0.83	1.11E−15	135.40%
	M4IP	4.67	8.83	13.25	36.13	0.84	−−	−−
5849	LO	10.24	21.52	36.01	2500	0.71	4.81E−57	145.68%
	Mean	9.82	20.84	35.99	1290.77	0.73	4.42E−01	59.22%
	KNN	20	42.27	71.83	247.61	0.62	1.22E−127	181.24%
	MICE	7.75	15.49	24.12	67.26	0.66	1.11E−15	144.05%
	MF	9.95	21.54	36.44	137.96	0.68	2.61E−294	140.13%
	BRITS	6.14	11.45	16.98	45.32	0.81	6.57E−04	85.32%
	GRU-D	5.2	9.68	14.55	35.63	0.83	1.87E−118	56.93%
	M4IP	3.39	6.17	9.05	25.7	0.85	−−	−−
25000	LO	8.66	18.15	29.31	2500	0.71	3.61E−83	98.43%
	Mean	2.46	5.25	8.94	329.95	0.77	6.57E−04	19.42%
	KNN	9.7	14.27	18.86	41.56	0.63	2.52E−195	72.38%
	MICE	8.99	13.08	17.18	36.48	0.67	9.88E−163	57.66%
	MF	8.26	13.03	17.86	41.23	0.7	8.99E−01	56.82%
	BRITS	7.23	15.31	25.76	106.09	0.82	2.60E−214	45.36%
	GRU-D	5.78	11.87	19.15	68.36	0.83	9.37E−251	33.29%
	M4IP	4.63	8.35	12.29	33.23	0.85	−−	−−

[1] @25%, @50%, @57%, Max: the distribution of all imputed values in terms of AE.
[2] PCC: Pearson correlation coefficient, indicating the correlations between imputed values and the corresponding ground truth.
[3] P: P-value of an imputation T-test with $\alpha = 0.05$.
[4] Imp/rovement: Imputation improvement between M4IP and the current-row method in terms of AE.

Table 6
Ablation study results of each component in M4IP.

Task	Metric	No-neaten	No-recurrent	No-estimate	No-constraint	Full
4019	MAE	9.56 ± 26.42	11.62 ± 56.80	10.96 ± 22.88	8.73 ± 6.79	8.73 ± 6.68
	MRE	0.18 ± 0.21	0.18 ± 0.39	0.18 ± 0.26	0.16 ± 0.16	0.16 ± 0.15
	MACC	0.8128	0.8257	0.8530	0. 9401	0.9402
41401	MAE	8.73 ± 28.09	9.34 ± 23.43	9.17 ± 12.01	8.73 ± 6.71	8.73 ± 6.68
	MRE	0.16 ± 0.17	0.18 ± 0.40	0.18 ± 0.16	0.16 ± 0.15	0.16 ± 0.15
	MACC	0.8475	0.8622	0.9127	0.9421	0.9421
5849	MAE	8.24 ± 9.06	9.46 ± 102.62	6.83 ± 24.05	6.13 ± 4.90	6.13 ± 4.32
	MRE	0.13 ± 0.12	0.19 ± 0.51	0.12 ± 0.14	0.12 ± 0.10	0.12 ± 0.08
	MACC	0.8544	0.8068	0.9077	0.9376	0.9378
2500	MAE	8.89 ± 21.23	12.19 ± 8.47	8.73 ± 12.40	8.29 ± 6.03	8.29 ± 5.88
	MRE	0.17 ± 0.44	0.19 ± 0.50	0.18 ± 0.19	0.17 ± 0.15	0.17 ± 0.14
	MACC	0.9094	0.8786	0.8870	0.9218	0.9227

No-neaten: eliminate the neaten layer. No-recurrent: eliminate the imputation recurrent layer. No-estimate: eliminate the estimated layer. No-constraint: eliminate the constraint layer. FULL: full DIMM framework. MACC: accuracy of mortality prediction.

mainly because the charting values when an ICU patient is in a normal state and those when the patient is in a serious state can be very different; the location-based methods cannot capture the change state instantaneously.

Then, we computed the Pearson correlation of the imputed values with the ground truth for the imputation methods. The Strength of Association (SoA) standard we used was 0.0–0.2 no, 0.2–0.4 weak, 0.4–0.6 medium, 0.6–0.8 large, and 0.8–1.0 strong. From Table 5, we can see that all the imputed values have large correlations with the ground truth and that the imputed values from M4IP have the strongest correlations.

Next, we compared the AE value (the lower, the better) between M4IP and other baselines. As shown in Table 5 (improvement), our method got 1.26%–63.8%, 1.6%–194.62%, 59.22%–181.24%, 19.42%–98.43%, improvement for task 4019, 41401, 5849, 25000, respectively. Besides, we calculated the improvement of MAE, MRE, and downstream classification accuracy. For MAE, 49.09% improvement in terms of mean, 84.20% improvement in terms of variance. For MRE, 53.55% improvement in terms of mean, 63.87% improvement in terms of variance. For classification accuracy, improved 10.42%.

Finally, to verify whether M4IP is a significant advancement compared with the baselines, we conducted a hypothesis test (T-test), with $\alpha = 0.05$ and H_0: no significant difference. The test results are shown in Table 5. From the results, we can see that all the p-values are smaller than α; thus, the hypothesis is rejected. Therefore, in the ICU missing value imputation scenario, our proposed M4IP is significantly better than the compared baseline methods.

6.5.1. Ablation study

To demonstrate the consistency of the proposed method, we conducted an ablation study in this section. The details are listed in Table 6. From the table, we find that the mean MAE and MRE are similar to those of the full M4IP, while the variance greatly increases when eliminating the neaten layer. This result occurs mainly because the neaten layer can facilitate the process of finding better initial imputation values. For the recurrent layer, when we destroy it, the imputation performance greatly decreases. This suggests that the recurrent layer is the main component in M4IP, providing the main driving force behind the missing value imputation. If we remove the estimated layer, compared to those of the full M4IP, both the mean and variance of the MAE increase (main: +5.30%, variance: +110.88%). This proves that adopting an FR estimation and an HR estimation is necessary. While the constraint layer provides less support in terms of the mean value of the MAE and MRE, it can help decrease the variance by approximately 1.65%. We can deduce from the table that each component in M4IP is necessary.

7. Conclusions

In this work, we proposed a novel 2-step model to address imputation problems in clinical prediction tasks. In the first step of M4IP, four factors, i.e., *masking*, *time interval*, *bursty*, and *cumulative*, are considered. In the second step, the imputed values are optimized through additional layers with the aim of exploiting clinical patterns and constraining the imputed values. To the best of our knowledge, M4IP is the first framework to consider the imputation quality when handling irregularly sampled, missing, or noisy data, which are common data quality issues in medical applications. The experimental results demonstrated that M4IP outperforms state-of-the-art methods in terms of both the imputation quality and classification performance. However, although our model achieves state-of-the-art performance on missing value imputation for ICU data, there is still much that can be done to boost the performance from a clinical perspective, such as domain knowledge-based imputation. Our future work will focus on this topic.

CRediT authorship contribution statement

Zhenkun Shi: Conceptualization Methodology and Writing original. **Sen Wang:** Conceptualization and Investigation. **Lin Yue:** Writing – review & editing. **Xianglin Zuo:** Investigation, and Validation. **Lixin Pang:** Writing – review & editing. **Wanli Zuo:** supervision. **Xue Li:** Supervision. All authors analyzed the results and revised the paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Tianjin Synthetic Biotechnology Innovation Capability Improvement Program (No. TSBICIP-CXRC-018), the Nature Science Foundation of Jilin Province (Nos. 20180101330JC and 20190302029GX), and the Scientific and Technological Development Program of Jilin Province (Nos. 20180520022JH and 20190302109GX). The authors also gratefully acknowledge the financial support from the China Scholarship Council (No. 201706170617).

References

[1] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, Y. Li, Brits: Bidirectional recurrent imputation for time series, in: Advances in Neural Information Processing Systems, 2018, pp. 6776–6786.
[2] H. Kang, The prevention and handling of the missing data, Kor. J. Anesthesiol. 64 (5) (2013) 402.
[3] A. Sharafoddini, J.A. Dubin, D.M. Maslove, J. Lee, A new insight into missing data in intensive care unit patient profiles: Observational study, JMIR Med. Inform. 7 (1) (2019).
[4] W.-C. Lin, C.-F. Tsai, Missing value imputation: a review and analysis of the literature (2006–2017), Artif. Intell. Rev. 53 (2) (2020) 1487–1509.
[5] S. Armijo-Olivo, S. Warren, D. Magee, Intention to treat analysis, compliance, drop-outs and how to deal with missing data in clinical research: a review, Phys. Therapy Rev. 14 (1) (2009) 36–49, https://doi.org/10.1179/174328809x405928.
[6] H.-F. Yu, N. Rao, I. S. Dhillon, Temporal regularized matrix factorization for high-dimensional time series prediction, in: Advances in neural information processing systems, 2016, pp. 847–855.
[7] Z. Shi, W. Zuo, W. Chen, L. Yue, J. Han, L. Feng, User relation prediction based on matrix factorization and hybrid particle swarm optimization, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 1335–1341.
[8] Z. Zhang, Multiple imputation with multivariate imputation by chained equation (mice) package, Ann. Transl. Med. 4 (2) (2016).
[9] D.M. Kreindler, C.J. Lumsden, The effects of the irregular sample and missing data in time series analysis, in: Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data, CRC Press, 2016, pp. 149–172.
[10] M. Soley-Bori, Dealing with missing data: Key assumptions and methods for applied analysis, Boston University, 2013.
[11] L. Bonomi, X. Jiang, A mortality study for icu patients using bursty medical events, in: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), IEEE, 2017, pp. 1533–1540.
[12] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (1) (2018) 6085.
[13] N. Liu, P. Lu, W. Zhang, J. Wang, Knowledge-aware deep dual networks for text-based mortality prediction, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE, 2019, pp. 1406–1417.
[14] Q. Ma, Y. Gu, W.-C. Lee, G. Yu, H. Liu, X. Wu, Remian: Real-time and error-tolerant missing value imputation, ACM Trans. Knowl. Discov. Data (TKDD) 14 (6) (2020) 1–38.
[15] G. Harari, M.S. Green, S. Zelber-Sagi, Estimation and development of 10-and 20-year cardiovascular mortality risk models in an industrial male workers database, Prevent. Med. 103 (2017) 26–32.
[16] A.E. Johnson, T.J. Pollard, L. Shen, H.L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, Mimic-iii, a freely accessible critical care database, Sci. Data 3 (2016) 160035.
[17] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, Z. Xu, Missing value estimation for mixed-attribute data sets, IEEE Trans. Knowl. Data Eng. 23 (1) (2010) 110–121.
[18] A. Vesin, E. Azoulay, S. Ruckly, L. Vignoud, K. Rusinovà, D. Benoit, M. Soares, P. Azeivedo-Maia, F. Abroug, J. Benbenishty, et al, Reporting and handling missing values in clinical studies in intensive care units, Intensive care Med. 39 (8) (2013) 1396–1404.
[19] L. Zhang, Y. Zhao, Z. Zhu, D. Shen, S. Ji, Multi-view missing data completion, IEEE Trans. Knowl. Data Eng. 30 (7) (2018) 1296–1309.
[20] W. Zhang, T. Luo, S. Qiu, J. Ye, D. Cai, X. He, J. Wang, Identifying genetic risk factors for alzheimer's disease via shared tree-guided feature learning across multiple tasks, IEEE Trans. Knowl. Data Eng. 30 (11) (2018) 2145–2156.
[21] S. Van Buuren, Flexible imputation of missing data, Chapman and Hall/CRC, 2018.
[22] I.B. Aydilek, A. Arslan, A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm, Inf. Sci. 233 (2013) 25–35.
[23] H. Verma, S. Kumar, An accurate missing data prediction method using lstm based deep learning for health care, in: Proceedings of the 20th International Conference on Distributed Computing and Networking, ACM, 2019, pp. 371–376.
[24] J. Yoon, W.R. Zame, M. van der Schaar, Estimating missing data in temporal data streams using multi-directional recurrent neural networks, IEEE Trans. Biomed. Eng. (2018).
[25] D. Mondal, D.B. Percival, Wavelet variance analysis for gappy time series, Ann. Inst. Stat. Math. 62 (5) (2010) 943–966.
[26] J. Tan, W. Liu, T. Wang, N. N. Xiong, H. Song, A. Liu, Z. Zeng, An adaptive collection scheme-based matrix completion for data gathering in energy-harvesting wireless sensor networks, IEEE Access (2019).
[27] Y. Chi, Y.M. Lu, Y. Chen, Nonconvex optimization meets low-rank matrix factorization: An overview, IEEE Trans. Signal Process. 67 (20) (2019) 5239–5269.
[28] M. Liu, L. Nie, X. Wang, Q. Tian, B. Chen, Online data organizer: micro-video categorization by structure-guided multimodal dictionary learning, IEEE Trans. Image Process. 28 (3) (2018) 1235–1247.
[29] X. Lin, P.C. Boutros, Optimization and expansion of non-negative matrix factorization, BMC Bioinf. 21 (1) (2020) 1–10.
[30] M.J. Azur, E.A. Stuart, C. Frangakis, P.J. Leaf, Multiple imputation by chained equations: what is it and how does it work?, Int J. Methods Psychiat. Res. 20 (1) (2011) 40–49.
[31] Z. C. Lipton, D. Kale, R. Wetzel, Directly modeling missing data in sequences with rnns: Improved classification of clinical time series, in: F. Doshi-Velez, J. Fackler, D. Kale, B. Wallace, J. Wiens (Eds.), Proceedings of the 1st Machine Learning for Healthcare Conference, Vol. 56 of Proceedings of Machine Learning Research, PMLR, Children's Hospital LA, Los Angeles, CA, USA, 2016, pp. 253–270. URL http://proceedings.mlr.press/v56/Lipton16.html.

[32] B. Liu, Y. Li, S. Ghosh, Z. Sun, K. Ng, J. Hu, Complication risk profiling in diabetes care: A bayesian multi-task and feature relationship learning approach, IEEE Trans. Knowl. Data Eng. (2019).
[33] Z. Shi, W. Zuo, S. Liang, X. Zuo, L. Yue, X. Li, Iddsam: an integrated disease diagnosis and severity assessment model for intensive care units, IEEE Access 8 (2020) 15423–15435.
[34] Z. Shi, W. Zuo, W. Chen, L. Yue, Y. Hao, S. Liang, Dmmam: Deep multi-source multi-task attention model for intensive care unit diagnosis, in: International Conference on Database Systems for Advanced Applications, Springer, 2019, pp. 53–69.
[35] W.A. Knaus, D.P. Wagner, E.A. Draper, J.E. Zimmerman, M. Bergner, P.G. Bastos, C.A. Sirio, D.J. Murphy, T. Lotring, A. Damiano, et al, The apache iii prognostic system: risk prediction of hospital mortality for critically iii hospitalized adults, Chest 100 (6) (1991) 1619–1636.
[36] J.-L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, L. Thijs, The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure, Intensive care Med. 22 (7) (1996) 707–710.
[37] R.P. Moreno, P.G. Metnitz, E. Almeida, B. Jordan, P. Bauer, R.A. Campos, G. Iapichino, D. Edbrooke, M. Capuzzo, J.-R. Le Gall, et al, Saps 3—from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at icu admission, Intensive care Med. 31 (10) (2005) 1345–1355.
[38] G.C. Siontis, I. Tzoulaki, J.P. Ioannidis, Predicting death: an empirical evaluation of predictive tools for mortality, Arch. Internal Med. 171 (19) (2011) 1721–1726.
[39] A. Rajkomar, E. Oren, K. Chen, A.M. Dai, N. Hajaj, M. Hardt, P.J. Liu, X. Liu, J. Marcus, M. Sun, et al, Scalable and accurate deep learning with electronic health records, npj Digital Med. 1 (1) (2018) 18.
[40] H. Song, D. Rajan, J. Thiagarajan, A. Spanias, Attend and diagnose: Clinical time series analysis using attention models, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, pp. 1–8.
[41] L. Zhou, G. Hripcsak, Temporal reasoning with medical data—a review with emphasis on medical natural language processing, J. Biomed. Inf. 40 (2) (2007) 183–202.
[42] K.-I. Goh, A.-L. Barabási, Burstiness and memory in complex systems, EPL (Europhys. Lett.) 81 (4) (2008) 48002.
[43] Y. Wang, R. Chen, J. Ghosh, J. C. Denny, A. Kho, Y. Chen, B. A. Malin, J. Sun, Rubik: Knowledge guided tensor factorization and completion for health data analytics, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1265–1274.
[44] A. Milewski, K.L. Ferguson, T.E. Terndrup, Comparison of pulmonary artery, rectal, and tympanic membrane temperatures in adult intensive care unit patients, Clin. Pediat. 30 (4_suppl) (1991) 13–16.
[45] D. Chemla, V. Castelain, M. Humbert, J.-L. Hébert, G. Simonneau, Y. Lecarpentier, P. Hervé, New formula for predicting mean pulmonary artery pressure using systolic pulmonary artery pressure, Chest 126 (4) (2004) 1313–1317.
[46] Y. Vodovotz, G. An, I.P. Androulakis, A systems engineering perspective on homeostasis and disease, Front. Bioeng. Biotechnol. 1 (2013) 6.
[47] L. Joseph, C. Reinhold, Introduction to probability theory and sampling distributions, Amer. J. Roentgenol. 180 (4) (2003) 917–923.
[48] S. Wang, X. Li, L. Yao, Q.Z. Sheng, G. Long, et al, Learning multiple diagnosis codes for icu patients with local disease correlation mining, ACM Trans. Knowl. Discov. Data (TKDD) 11 (3) (2017) 31.
[49] S. Zhang, Nearest neighbor selection for iteratively knn imputation, J. Syst. Softw. 85 (11) (2012) 2541–2552.