HHS Public Access

Author manuscript

Proc (IEEE Int Conf Healthc Inform). Author manuscript; available in PMC 2024 December 26.

Published in final edited form as:

Proc (IEEE Int Conf Healthc Inform). 2024 June ; 2024: 135–140. doi:10.1109/ichi61247.2024.00025.

Multi-Task Deep Neural Networks for Irregularly Sampled Multivariate Clinical Time Series

Yuxi Liu[*], **Zhenhao Zhang**[†], **Shaowen Qin**[*], **Jiang Bian**[‡]

[*]College of Science and Engineering, Flinders University, Adelaide, SA, Australia

[†]College of Life Sciences, Northwest A&F University, Yangling, Shaanxi, China

[‡]College of Medicine, University of Florida, Gainesville, FL, USA

Abstract

Multivariate clinical time series data, such as those contained in Electronic Health Records (EHR), often exhibit high levels of irregularity, notably, many missing values and varying time intervals. Existing methods usually construct deep neural network architectures that combine recurrent neural networks and time decay mechanisms to model variable correlations, impute missing values, and capture the impact of varying time intervals. The complete data matrices thus obtained from the imputation task are used for downstream risk prediction tasks. This study aims to achieve more desirable imputation and prediction accuracy by performing both tasks simultaneously. We present a new multi-task deep neural network that incorporates the imputation task as an auxiliary task while performing risk prediction tasks. We validate the method on clinical time series imputation and in-hospital mortality prediction tasks using two publicly available EHR databases. The experimental results show that our method outperforms state-of-the-art imputation-prediction methods by significant margins. The results also empirically demonstrate that the incorporation of time decay mechanisms is a critical factor for superior imputation and prediction performance. The novel deep imputation-prediction network proposed in this study provides more accurate imputation and prediction results with EHR data. Future work should focus on developing more effective time decay mechanisms for simultaneously enhancing the imputation and prediction performance of multi-task learning models.

Index Terms—

Electronic Health Record; Multi-Task Learning; Temporal Representation Learning; Irregularly Sampled Multivariate Time Series

I. Introduction

Digital health systems are widely available and being integrated into routine healthcare operations, resulting in growth in electronic health records (EHRs) data. With advances in data processing tools and methods, there has been an increased interest in establishing health risk prediction models as a key instrument in clinical decision support.

liu1356@flinders.edu.au .

However, EHR data has its unique characteristics, such as high dimensionality, sparsity, irregularity, temporality, bias, etc. It is technically challenging to apply traditional machine learning or statistical models to such data. The high degree of irregularity, many missing values and varying time intervals, needs to be dealt with when establishing predictive models. EHR data irregularity is a natural consequence of health care provision, as every patient is different. For example, patients are more likely to be examined by healthcare specialists when changes in their health status or treatment decisions occur, hence the intervals between physiological variables are often irregular. Additionally, the variation of missing data patterns adds another layer of complexity, which would affect the performance of downstream risk prediction.

Most of the previous research studies with EHR data have been focused on the provision of deep imputation prediction models [1]–[6]. There are three modes of imputation-prediction processing, each has its drawbacks. The first is to consider imputation and prediction as two separable steps [3], [4], [6]–[9]. Although promising prediction performance has been demonstrated, these prediction models have not attempted to learn the impact of the patterns of missing data in EHR data. This may lead to suboptimal prediction performance. As a better alternative, imputation and prediction can be tuned together within an end-to-end learning framework rather than be separated into two parts. This is the second mode. Despite its efficacy, existing architectures for such modes are specifically proposed for improving risk prediction performance [1], [5], [10], [11]. When used for imputation and prediction tasks, the architecture treats both as separate optimization tasks, which essentially is not different from the first mode. The third imputation-prediction processing mode is similar to that used by the second, with the difference that the objective of the third is to simultaneously perform imputation and prediction tasks [2], [12]–[15]. However, imputation and prediction tasks may lead to competition due to the shared parameter problem, as illustrated during multi-task learning for optimization in some studies [16]–[18]. This kind of optimization could also lead to suboptimal imputation and prediction results.

This study proposes to construct a single deep learning framework based on multi-task learning that performs the risk prediction task while incorporating the imputation task as an auxiliary task. The benefit of implementing the imputation task as an auxiliary task is that such an approach can improve risk prediction performance rather than competing with it. It is a novel multi-task deep neural network in which imputation and prediction tasks are implemented with an auxiliary network and a main network, respectively (Figure 1). The intuition behind our network architecture is that the direction of information flow is from the auxiliary network to the main network only. By doing so, the forward pass of the main network depends on the auxiliary network, while the inference of the auxiliary network does not depend on the main network. Therefore, imputation and prediction tasks can be implemented simultaneously within a single deep learning framework without competition. Our major contributions are provided as follows:

- We present a new multi-task deep neural network to simultaneously carry out imputation and prediction tasks using irregularly sampled multivariate clinical time series.

- To our knowledge, this is the first research to perform risk prediction tasks by incorporating the imputation task as an auxiliary task while carrying out both simultaneously.
- Experiments on data from two real-world EHR databases using our proposed method demonstrate superior prediction and imputation accuracy.

II. Related Work

A. Deep Neural Networks for Missing Data Imputation

The BRITS [2] employs a bidirectional recurrent neural network (RNN) to impute missing values in multivariate time series data and then exploits these imputed values to predict the final imputed values. The V-RIN [14] integrates a Variational autoencoder and a GRU into a single framework. The incorporated Variational autoencoder uses an encoder network to learn the distribution of multivariate clinical time series data and a decoder network to generate the reconstructed data distribution where the reconstructed values are the imputed values. The GRU used in V-RIN continues to be a recurrent imputation network, which aims at capturing the variation pattern of input variables at the time dimension.

Towards the generative adversarial networks (GAN) architecture, GRUI-GAN [3], E^{2}GAN [4], and STING [6] are representative GAN based imputation methods. They take the vector of actual samples as input to the GAN architecture in which a generator generates the imputed values for missing values, and a discriminator distinguishes the imputed values from the actual values. Accordingly, the generator and discriminator can lead to competition, which is the rationale behind the GAN architecture.

Deep neural networks with attention mechanisms (also known as attention-based neural networks) have been applied for the imputation of missing data. The study by [9] proposes a deep imputation method (shorten for MTSIT) based on a combination of Transformer encoder [19] and linear decoder. The study by [15] proposes a deep imputation-prediction method (shorten for MIAM) based on the self-attention mechanism [19], which focuses on the provision of multivariate clinical time series missing data imputation.

III. Method

A. Data Representation

We represent a multivariate clinical time series with up to K physiological variables as $X = \{x_1, x_2, \cdots, x_T\} \in \mathbb{R}^{K \times T}$, where T is the number of medical records. Since X can be incomplete, we represent the missing values in x_t^k by introducing a masking vector M_t^k as:

$$M_t^k = \begin{cases} 1, \ if \ \ x_t^k \ \ is \ \ observed \\ 0, \ otherwise \end{cases}. \tag{1}$$

Let s_t represent the timestamp when the t-th medical record is obtained, and Δ_t represent the time interval for each physiological variable since its last medical record. The Δ_t^k can be written as:

$$\Delta_t^k = \begin{cases} s_t - s_{t-1} + \Delta_{t-1}^k, & t > 1, M_{t-1}^k = 0 \\ s_t - s_{t-1}, & t > 1, M_{t-1}^k = 1 \\ 0, & t = 1 \end{cases}. \tag{2}$$

Let $D = \left\{\left(X_n, Y_n^{(I)}, Y_n^{(P)}\right) \mid n = 1, 2, \cdots, N\right\}$ represent an EHR dataset with up to N multivariate clinical time series. Each has two target labels $Y_n^{(I)}$ and $Y_n^{(P)}$ for imputation and prediction tasks.

B. Network Architecture

1) Convolutional Component: Given multivariate clinical time series X, we first construct a learnable variable to carry out prefilling operations. Let ψ represent a learnable variable, which is initialized as $\widetilde{X} = M \cdot X + (1 - M) \cdot \psi$. We then apply the zero vector padding to $\widetilde{X}$ by embedding a zero vector before the first record of $\widetilde{X}$ and after the last record of $\widetilde{X}$. We finally feed $\widetilde{X}$ into a convolutional component.

In particular, a combination of up to K kernels $\{W_k\}_{k=1}^K$ is applied to the corresponding K variables. For example, $\widetilde{x}_{t:t+l-1}^k$ represents the concatenation of k-th variable of different records $\left\{\widetilde{x}_t^k, \widetilde{x}_{t+1}^k, \cdots, \widetilde{x}_{t+l-1}^k\right\}$. A kernel $W_k \in \mathbb{R}^l$ is applied to the window of $\widetilde{x}_{t:t+l-1}^k$ to generate a new latent variable $v_t^k \in \mathbb{R}$ with ReLU activation function as:

$$v_t^k = ReLU\left(\widetilde{x}_{t:t+l-1}^k \cdot W_k + b_k\right), \tag{3}$$

where $ReLU(x) = max(x, 0)$ and $b_k \in \mathbb{R}$ is a bias. In a follow-up step, W_k is implemented as a sliding window in order to generate a latent vector $v^k = \left\{v_1^k, v_2^k, \cdots, v_T^k\right\}$. The final representation of $\widetilde{X}$ can be $v \in \mathbb{R}^{K \times T}$ based on concatenating all of those latent vectors.

2) Residual Recurrent Component: The residual recurrent component is built upon GRU [20]. The GRU is characterized by the reset gate r_t and the update gate u_t, which decide the information from the previous hidden state h_{t-1} should be updated or reset the previous hidden state h_{t-1} whenever needed. Given the final representation v_t obtained from the convolutional component, GRU generates h_t by the use of a linear combination of the previous hidden state h_{t-1} and the candidate state $\widetilde{h}_t$ as:

$$\begin{aligned} h_t = GRU(v_t) &= u_t \odot \widetilde{h}_t + (1 - u_t) \odot h_{t-1}, \\ u_t &= \sigma\left(W_u^1 \cdot h_{t-1} + W_u^2 \cdot v_t + b_u\right), \\ \widetilde{h}_t &= tanh\left(W_h^1 \cdot (r_t \odot h_{t-1}) + W_h^2 \cdot v_t + b_h\right), \\ r_t &= \sigma\left(W_r^1 \cdot h_{t-1} + W_r^2 \cdot v_t + b_r\right), \end{aligned}$$

(4)

where ⊙ is the element-wise multiplication, and σ is the sigmoid function. The u_t controls the information from the previous hidden state h_{t-1} and the candidate state $\widetilde{h}_t$. The r_t decides the proper amount of information from the previous hidden state h_{t-1} that contributes to $\widetilde{h}_t$ generation. We forward an identity mapping of the GRU input to its output side as $h_t^{'} = ResGRU(v_t) = GRU(v_t) + v_t$.

3) Time Decay Mechanism: To capture the impact of varying time intervals, competitive time decay mechanisms that fit a deep imputation-prediction network are sought and critically reviewed. Collectively, we separately incorporate three types of time decay mechanisms [1], [5], [21] into the proposed network architecture to test their efficacy on imputation and prediction performance. We augment the residual recurrent component with time decay mechanisms [1], [5] respectively. The mathematical formulations for [1], [5] are as:

$$f_1(\Delta_t) = exp\{-max(0, W_\gamma \cdot \Delta_t + b_\gamma)\},$$

(5)

$$\begin{aligned} f_2(\Delta_t) &= \frac{1}{log(e + \Delta_t)}, \\ f_3(\Delta_t) &= e^{-\Delta_t}, \\ f_4(\Delta_t) &= \frac{1}{\Delta_t}, \end{aligned}$$

(6)

where $f_2(\cdot)$, $f_3(\cdot)$, and $f_4(\cdot)$ are three types of decay functions. The above $f(\cdot)$ functions are integrated into the GRU architecture that contribute hidden state representation generation. Accordingly, $\hat{h}_{t-1}$ can be written as $f(\Delta_t) \odot h_{t-1}$. Subsequently, Eq. (4) can be rewritten as:

$$\begin{aligned} h_t = GRU(v_t) &= u_t \odot \widetilde{h}_t + (1 - u_t) \odot \hat{h}_{t-1}, \\ u_t &= \sigma\left(W_u^1 \cdot \hat{h}_{t-1} + W_u^2 \cdot v_t + b_u\right), \\ \widetilde{h}_t &= tanh\left(W_h^1 \cdot \left(r_t \odot \hat{h}_{t-1}\right) + W_h^2 \cdot v_t + b_h\right), \\ r_t &= \sigma\left(W_r^1 \cdot \hat{h}_{t-1} + W_r^2 \cdot v_t + b_r\right). \end{aligned}$$

(7)

Compared with time decay mechanisms in [1] and [5], [21] also takes the similarity between medical records into consideration on the time decay mechanism. In other words, if the similarity between two medical records is significant, the importance of the previous one should be slightly decayed. This is achieved by combining the attention function [19] and the decay function $\frac{1}{log(e + \Delta_t)}$. The mathematical formulations for [21] are as:

$$
\begin{aligned}
Q_T^k &= W_Q^k \cdot \widetilde{x}_T^k, \\
K_t^k &= W_K^k \cdot \widetilde{x}_t^k, \\
\eta_t^k &= tanh\left(\frac{Q_T^k \cdot K_t^k}{\beta_k \cdot log\left(e + \left(1 - \sigma \cdot \left(Q_T^k \cdot K_t^k\right)\right) \cdot \Delta_t\right)}\right), \\
\alpha &= Softmax(\eta), \\
\widetilde{X}' &= \alpha \odot \widetilde{X}.
\end{aligned}
\tag{8}
$$

4) Multi-Task Learning for Imputation and Prediction Tasks: Multi-task learning is a single shared machine learning model that performs multiple target tasks simultaneously. Multi-task learning with deep neural networks can be done using either hard or soft parameter sharing of hidden layers. The former allows target tasks to share parameters from a series of hidden layers, while the latter allows each target task to have its own backbone with its own parameters. Previous studies suggest that multiple target tasks lead to competition regardless of the hard or soft parameter-sharing methods. In response to the competition, we construct different optimizers for imputation and prediction tasks and then perform the risk prediction task by incorporating the imputation task as an auxiliary task. As Figure 1 shows, an auxiliary network and a main network are developed and introduced to the imputation and prediction tasks. The key aspect of our network architecture is that the direction of information flow is from the auxiliary network to the main network only. Accordingly, the forward pass of the main network depends on the auxiliary network, while the inference of the auxiliary network does not depend on the main network. Because of this, imputation and prediction tasks can be implemented simultaneously within a single deep learning framework without competition.

Now we define the objective functions for the imputation and prediction tasks. Given the final representation h', we utilize a fully connected layer to impute missing values as:

$$
\hat{y}^{(I)} = W_y^{(I)} \cdot h' + b_y^{(I)}.
\tag{9}
$$

The mean square error (MAE) for the imputation task as:

$$
\mathscr{L}^{(I)} = \frac{1}{N} \sum_{n=1}^{N} \left(M_n \odot \hat{y}_n^{(I)} - M_n \odot Y_n^{(I)}\right)^2.
\tag{10}
$$

For the risk prediction task, we utilize $h_T^{'}$ as input for a Softmax output layer in order to obtain the predicted $\hat{y}^{(p)}$ as:

$$
\hat{y}^{(P)} = Softmax\left(W_y^{(P)} \cdot h_T^{'} + b_y^{(P)}\right).
\tag{11}
$$

The average cross-entropy with a constraint L-infinity norm $\|\cdot\|_\infty$ for the risk prediction task as:

$$\mathscr{L} = -\frac{1}{N}\sum_{n=1}^{N}\left(\left(Y_n^{(P)}\right)^\top \cdot log\left(\hat{y}_n^{(P)}\right) + \left(1-Y_n^{(P)}\right)^\top \cdot log\left(1-\hat{y}_n^{(P)}\right)\right),$$

$$\mathscr{L}^{(P)} = \mathscr{L} + \lambda \cdot \|\theta - \phi\|_\infty,$$

$$\|\theta - \phi\|_\infty = \lim_{p \to \infty}\left(\sum_{j}^{J}|\theta_j - \phi_j|^p\right)^{\frac{1}{p}}, \tag{12}$$

where λ is a scaling parameter that handles the contribution of cross-entropy and constraint, $\|\theta - \phi\|_\infty$ is the distance between the auxiliary network parameter $\{\theta_j\}_{j=1}^{J}$ and the main network parameter $\{\phi_j\}_{j=1}^{J}$, and J is the number of shared layers in the network architecture.

IV. Experiments

A. Datasets, Tasks, and Evaluation metrics

We extract 21,105 and 36,670 patients/samples from the MIMIC-III and eICU databases, where the Positive (likely to die)/Negative (unlikely to die) ratios are 1:6.56 and 1:7.49, respectively. We assess the imputation performance using MAE and MRE (between predicted and actual values) and the prediction performance using AUROC and AUPRC. The data extraction, code, and statistics of physiological variables are available in https://github.com/LZlab01/MultiTaskEHR.

B. Baselines

We utilize GRU-D [1], BRITS [2], V-RIN [14], GRU-IGAN [3], E^{2}GAN [4], STING [6], MTSIT [9], MIAM [15] as baselines for comparison. We provide six variants of our approach as:

Ours$_\alpha$: we incorporate the time decay mechanism [1] into the residual recurrent component.

Ours$_\beta$: we incorporate the time decay mechanism [5] (i.e., the first row of Eq. (6)) into the residual recurrent component.

Ours$_\gamma$: we incorporate the time decay mechanism [5] (i.e., the second row of Eq. (6)) into the residual recurrent component.

Ours$_\delta$: we incorporate the time decay mechanism [5] (i.e., the third row of Eq. (6)) into the residual recurrent component.

Ours$_\epsilon$: we incorporate the time decay mechanism [21] into the network architecture.

Ours$_\varepsilon$: we do not perform any time decay mechanism.

C. Implementation details

The training was done in a machine equipped with a CPU: AMD EPYC 7543, 80GB RAM, and a GPU: NVIDIA A40 with 48GB of memory using Pytorch 1.10.0. We randomly used 70%, 15%, and 15% of the dataset as training, validation, and testing sets. For the MIMIC-III dataset, the number of physiological variables K is 17. For the convolutional component, the kernel size is 3 and the stride is 1. For the residual recurrent component, the dimension of hidden variables is 17. For multi-task learning, the scaling parameter λ is 0.002, and the learning rates for the imputation and prediction optimizers are 0.0065 and 0.0034. For the eICU dataset, the number of physiological variables K is 16. For the convolutional component, the kernel size is 3 and the stride is 1. For the residual recurrent component, the dimension of hidden variables is 16. For multi-task learning, the scaling parameter λ is 0.0013, and the learning rates for the imputation and prediction optimizers are 0.0077 and 0.0022. We integrated the regression component [2] into GRU-D network architecture to generate imputation results. We replaced the linear decoder of MTSIT with a Softmax output layer to generate prediction results. For a fair comparison, we used the complete data matrices imputed by GRUI-GAN, E^{2}GAN, STING as input to GRU to generate prediction results.

V. Performance Evaluation

As Table I shows, our method achieves the best imputation and prediction accuracy. Comparing the prediction results, it can be seen that Ours_{α} significantly and consistently outperforms $\text{Ours}_{\varepsilon}$ (i.e., without any time decay mechanism). These results suggest that the time decay mechanism [1] plays an important role in addressing varying time intervals of multivariate time series data, which leads to good prediction performance. Ours_{α} also consistently outperforms other baseline methods in the risk prediction task. These results suggest that the time decay mechanism [1] is particularly well suited for improving the downstream risk prediction performance of our network architecture. Moreover, the three variant methods (Ours_{β}, Ours_{γ}, Ours_{δ}) consistently outperform $\text{Ours}_{\varepsilon}$. These results suggest that capturing the effect of varying time intervals can help improve imputation performance.

VI. Conclusion

We present a new multi-task deep neural network that performs risk prediction tasks by incorporating the imputation task as an auxiliary task. We experimentally demonstrate that the proposed method achieves the best imputation and prediction accuracy by conducting imputation and prediction experiments on two well-known EHR datasets. Moreover, we empirically demonstrate that the incorporation of time decay mechanisms is a critical factor for superior imputation and prediction performance. However, incorporating existing time decay mechanisms into our network architecture suffers some limitations. For example, incorporating the time decay mechanism described in [1] achieves the best prediction but suboptimal imputation performance. In contrast, incorporating the time decay mechanism proposed by [5] into our network architecture achieves the best imputation but poorer prediction performance. Further modeling work will have to be conducted in order to

develop more effective time decay mechanisms for simultaneously enhancing the imputation and prediction performance of multi-task learning models.

Acknowledgement

This project is in part supported by NIH grants R01AG083039, RF1AG084178, R01AG084236, R01AI172875, R01AG080624, UL1TR001427, and CDC U18 DP006512.

References

[1]. Che Z, Purushotham S, Cho K, Sontag D, and Liu Y, "Recurrent neural networks for multivariate time series with missing values," Scientific reports, vol. 8, no. 1, pp. 1–12, 2018. [PubMed: 29311619]

[2]. Cao W, Wang D, Li J, Zhou H, Li L, and Li Y, "Brits: Bidirectional recurrent imputation for time series," Advances in neural information processing systems, vol. 31, 2018.

[3]. Luo Y, Cai X, Zhang Y, Xu J et al. , "Multivariate time series imputation with generative adversarial networks," Advances in neural information processing systems, vol. 31, 2018.

[4]. Luo Y, Zhang Y, Cai X, and Yuan X, "E2gan: End-to-end generative adversarial network for multivariate time series imputation," in Proceedings of the 28th international joint conference on artificial intelligence. AAAI Press, 2019, pp. 3094–3100.

[5]. Tan Q, Ye M, Yang B, Liu S, Ma AJ, Yip TC-F, Wong GL-H, and Yuen P, "Data-gru: Dual-attention time-aware gated recurrent unit for irregular multivariate time series," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 930–937.

[6]. Oh E, Kim T, Ji Y, and Khyalia S, "Sting: Self-attention based time-series imputation networks using gan," in 2021 IEEE International Conference on Data Mining (ICDM). IEEE, 2021, pp. 1264–1269.

[7]. Xu D, Sheng JQ, Hu PJ-H, Huang T-S, and Hsu C-C, "A deep learning–based unsupervised method to impute missing values in patient records for improved management of cardiovascular patients," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 6, pp. 2260–2272, 2020.

[8]. Yang S, Dong M, Wang Y, and Xu C, "Adversarial recurrent time series imputation," IEEE Transactions on Neural Networks and Learning Systems, 2020.

[9]. Yıldız AY, Koç E, and Koç A, "Multivariate time series imputation with transformers," IEEE Signal Processing Letters, vol. 29, pp. 2517–2521, 2022.

[10]. Mouselinos S, Polymenakos K, Nikitakis A, and Kyriakopoulos K, "Main: multihead-attention imputation networks," in 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021, pp. 1–8.

[11]. Liu Y, Qin S, Zhang Z, and Shao W, "Compound density networks for risk prediction using electronic health records," in 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2022, pp. 1078–1085.

[12]. Shen L, Ma Q, and Li S, "End-to-end time series imputation via residual short paths," in Asian conference on machine learning. PMLR, 2018, pp. 248–263.

[13]. Shukla SN and Marlin BM, "Interpolation-prediction networks for irregularly sampled time series," arXiv preprint arXiv:1909.07782, 2019.

[14]. Mulyadi AW, Jun E, and Suk H-I, "Uncertainty-aware variational-recurrent imputation network for clinical time series," IEEE Transactions on Cybernetics, vol. 52, no. 9, pp. 9684–9694, 2021.

[15]. Lee Y, Jun E, Choi J, and Suk H-I, "Multi-view integrative attention-based deep representation learning for irregular clinical time-series data," IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 8, pp. 4270–4280, 2022. [PubMed: 35511839]

[16]. Fifty C, Amid E, Zhao Z, Yu T, Anil R, and Finn C, "Efficiently identifying task groupings for multi-task learning," Advances in Neural Information Processing Systems, vol. 34, pp. 27 503–27 516, 2021.

[17]. Mueller D, Andrews N, and Dredze M, “Do text-to-text multi-task learners suffer from task conflict?” arXiv preprint arXiv:2212.06645, 2022.

[18]. Sumbul G and Demir B, “Plasticity-stability preserving multi-task learning for remote sensing image retrieval,” IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–16, 2022.

[19]. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, and Polosukhin I, “Attention is all you need,” in Advances in neural information processing systems, 2017, pp. 5998–6008.

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

[21]. Ma L, Zhang C, Wang Y, Ruan W, Wang J, Tang W, Ma X, Gao X, and Gao J, “Concare: Personalized clinical feature embedding via capturing the healthcare context,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 833–840.

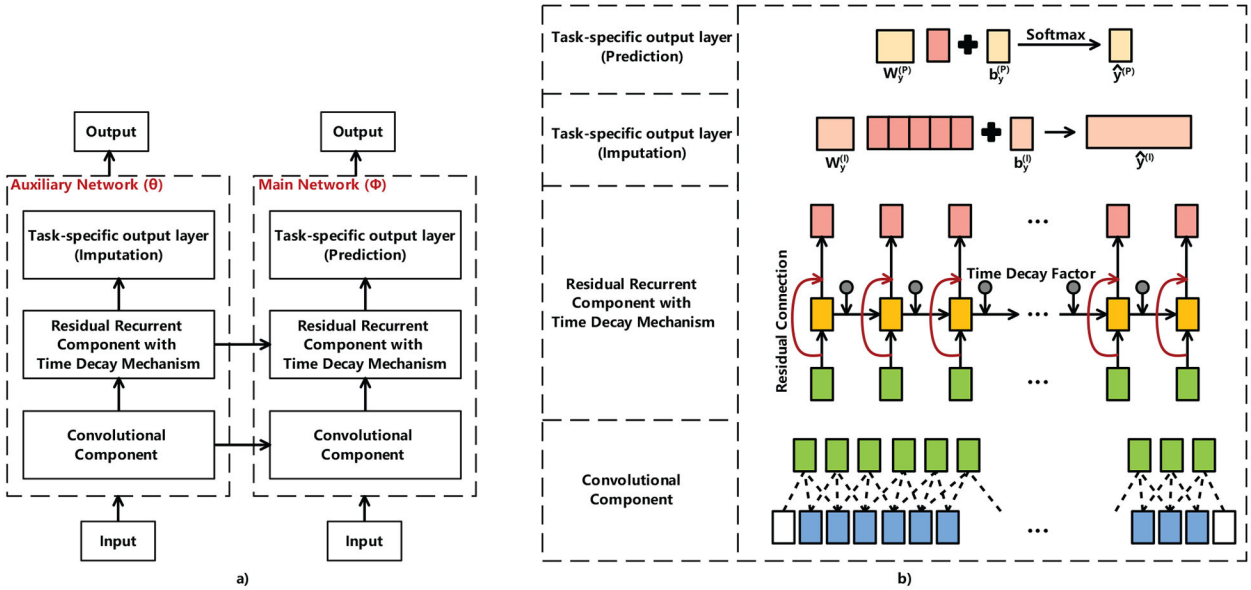

Fig. 1.
Schematic description of the proposed approach.

Proc (IEEE Int Conf Healthc Inform). Author manuscript; available in PMC 2024 December 26.

TABLE I

Performance of all methods on imputation and prediction tasks. The prediction windows are set to the first 24 and 48 hours after admission.

Method	Imputation		Prediction	
MIMIC-III/48 hours	**MAE**	**MRE**	**AUROC**	**AUPRC**
GRU-D	3.6873(0.0218)	36.20%(0.0021)	0.7294(0.0097)	0.2771(0.0156)
BRITS	5.3631(0.3804)	52.65%(0.0374)	0.7447(0.0092)	0.2879(0.0168)
V-RIN	3.1522(0.0080)	31.15%(0.0010)	0.7758(0.0003)	0.3244(0.0010)
GRUI-GAN	7.1359(0.0055)	70.05%(0.0005)	0.7619(0.0077)	0.3349(0.0178)
E^2GAN	6.9705(0.0104)	68.43%(0.0010)	0.7652(0.0054)	0.3599(0.0133)
STING	5.1522(0.0202)	50.88%(0.0020)	0.7667(0.0106)	0.3402(0.0187)
MTSIT	1.6965(0.1114)	21.16%(0.0139)	0.6841(0.0171)	0.2584(0.0182)
MIAM	2.0941(0.0596)	26.13%(0.0074)	0.7192(0.0158)	0.2600(0.0111)
Ours_{α}	0.6017(0.0289)	7.50%(0.0036)	**0.8031(0.0045)**	**0.3800(0.0126)**
Ours_{β}	0.4246(0.0600)	5.29%(0.0074)	0.7420(0.0435)	0.2982(0.0438)
Ours_{γ}	0.4080(0.0531)	5.09%(0.0066)	0.7331(0.0108)	0.2689(0.0145)
Ours_{δ}	**0.3743(0.0565)**	**4.66%(0.0070)**	0.7093(0.0258)	0.2547(0.0271)
Ours_{ϵ}	1.3653(0.0232)	17.03%(0.0028)	0.7754(0.0085)	0.3621(0.0176)
Ours_{ϵ}	0.8525(0.0396)	10.63%(0.0049)	0.7789(0.0077)	0.3553(0.0145)
Method	**Imputation**		**Prediction**	
eICU/48 hours	**MAE**	**MRE**	**AUROC**	**AUPRC**
GRU-D	2.8066(0.0107)	21.43%(0.0008)	0.7195(0.0111)	0.2631(0.0145)
BRITS	4.0963(0.3359)	31.26%(0.0257)	0.7254(0.0057)	0.2573(0.0062)
V-RIN	1.8357(0.1097)	14.01%(0.0083)	0.7846(0.0139)	0.3373(0.0117)
GRUI-GAN	9.9809(0.0056)	76.26%(0.0002)	0.7280(0.0105)	0.2871(0.0120)
E^2GAN	9.7912(0.0111)	74.70%(0.0006)	0.7294(0.0106)	0.2970(0.0133)
STING	8.0315(0.0466)	61.21%(0.0036)	0.7475(0.0186)	0.2838(0.0197)
MTSIT	2.8713(0.1357)	21.92%(0.0103)	0.7237(0.0042)	0.2952(0.0103)
MIAM	2.2828(0.1288)	17.42%(0.0098)	0.7222(0.0099)	0.2513(0.0087)
Ours_{α}	0.8846(0.0697)	6.75%(0.0053)	**0.7984(0.0055)**	**0.3510(0.0075)**
Ours_{β}	**0.5852(0.0787)**	**4.46%(0.0060)**	0.7169(0.0155)	0.2650(0.0168)

Ours$_{\beta}$	0.3846(0.0687)	4.99%(0.0089)	0.6817(0.0145)	0.2341(0.0133)
Ours$_{\gamma}$	**0.3603(0.0561)**	**4.67%(0.0072)**	0.6766(0.0065)	0.2242(0.0086)
Ours$_{\delta}$	0.3913(0.0718)	5.07%(0.0093)	0.6785(0.0048)	0.2323(0.0040)
Ours$_{\epsilon}$	1.0752(0.0413)	13.95%(0.0053)	0.7422(0.0033)	0.3005(0.0077)
Ours$_{\varepsilon}$	0.7439(0.0472)	9.65%(0.0061)	0.7223(0.0075)	0.2847(0.0095)