Exploring Clinical Time Series Forecasting with Meta-Features in Variational Recurrent Models

Sibghat Ullah*, Zhao Xu†, Hao Wang‡, Stefan Menzel§, Bernhard Sendhoff§, and Thomas Bäck*

*Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands
Email: {s.ullah,t.h.w.baeck}@liacs.leidenuniv.nl

†NEC Laboratories Europe GmbH, Heidelberg, Germany
Email: zhao.xu@neclab.eu

‡Sorbonne Université, CNRS, LIP6, Paris, France
Email: hao.wang@lip6.fr

§Honda Research Institute Europe GmbH (HRI-EU), Offenbach/Main, Germany
Email: {stefan.menzel,bernhard.sendhoff}@honda-ri.de

***Abstract*—Clinical time series are known for irregular, highly-sporadic and strongly-complex structures and are consequently difficult to model by traditional state-space models. In this paper, we investigate the potential of applying variational recurrent neural networks (VRNNs) for forecasting clinical time series extracted from electronic health records (EHRs) of patients. Variational recurrent neural networks (VRNNs) combine recurrent neural networks (RNNs) and variational inference (VI) and are state-of-the-art methods to model highly-variable sequential data such as text, speech, time series and multimedia signals in a generative fashion. We propose to incorporate multiple correlated time series to improve the forecasting of VRNNs. The selection of these correlated time series is based on the similarity of the supplementary medical information e.g., disease diagnostics, ethnicity and age etc. between the patients. We evaluate the effectiveness of utilizing such supplementary information with root mean square error (RMSE), on clinical benchmark data-set "Medical Information Mart for Intensive Care (MIMIC III)" for multi-step-ahead prediction. We further perform subjective analysis to highlight the effects of the similarity of the supplementary medical information on individual temporal features e.g., Systolic Blood Pressure (SBP), Heart Rate (HR) etc. of the patients from the same data-set. Our results clearly show that incorporating the correlated time series based on the supplementary medical information can help improving the accuracy of the VRNNs for clinical time series forecasting.**

***Keywords*—time series forecasting, recurrent neural networks, deep-latent variable models, MIMIC III, Clinical Applications**

I. Introduction

Health care is one of the most exciting and challenging areas of machine learning and data mining. The use of electronic health records (EHRs) can significantly improve the existing health care systems since it can help identify early triage and risk evaluations [1]–[4] for certain group of patients at very early stages of treatment. Most of the existing electronic health records (EHRs) capture the temporal features for patients during their Intensive Care Unit (ICU) stay. Examples of such features include Heart Rate (HR), Oxygen Saturation Level (OSL), Body Temperature (BT) and Mean Blood Pressure (MBP) [5]. The set of these temporal signals can be used for further useful analysis such as phenotype classification [6]–[8], length-of-stay prediction [9], [10], risk-of-mortality prediction [10] and forecasting such signals for future time-steps. Unfortunately, these multivariate time series are characterized by highly-irregular[**], sporadic [11] and complex structures [12] and are consequently difficult to model by traditional methods.

Deep learning [13] has previously been applied to model medical signals[††]extracted from ICUs [14], [15]. Earlier studies to model such clinical signals however focused on tasks such as binary classification for length-of-stay (LOS) [9] (i.e., to identify the patients expected to stay longer in the ICU), phenotype classification [6]–[8], and survival analysis [16], [17]. It is important to state that deep learning in these earlier studies almost always focused on discriminative (a.k.a. conditional) feed-forward neural networks (FFNNs) and long short-term memory (LSTM) based recurrent neural networks (RNNs) with no or limited stochasticity.

In this paper, we limit ourselves to clinical time series forecasting. In the past, time series forecasting relied heavily on state-space models which are typically linear and are suited for univariate time series, although multivariate non-linear extensions of such models exist [18]. These methods require

[**] Irregular and sporadic multivariate time series in this context refers to a time series where the time intervals are not uniform and only a small subset of temporal features is observed at each time-step.

[††] The terms "temporal signals", "temporal features", "medical signals" and "time series" have been used interchangeably throughout the paper to refer to the same concept, i.e., time indexed variables of an individual patient which are observed in the ICU. In addition, terms "supplementary domain information", "supplementary medical information", "extra domain information" and "extra medical information" have also been used interchangeably throughout the paper to refer to the non-temporal subjective information about the patients, which is observed when the patient is admitted to the hospital or ICU.

specifications of trends, seasonality, cyclical effects and shocks in time series forecasting. As a result, these methods have higher interpretability. However, this interpretability (usually) comes at the cost of the prediction accuracy since such models lack the dynamic and complex nature of the multivariate time series extracted from modern ubiquitous systems such as economic transaction processing systems and electronic health record systems.

With the advent of deep learning, many methodologies have been proposed to employ recurrent neural networks (RNNs) for time series forecasting since RNNs are a natural choice for modelling sequential data-sets. Hybrid approaches to combine state-space models and deep learning have also been proposed [19]. Vanilla RNNs however have deterministic hidden states and lack the intrinsic stochasticity found in the latent variable models such as Hidden Markov Models (HMMs) and Kalman Filters. Recently, it has been argued [20]–[22] to incorporate some stochasticity in RNNs while modelling complex sequences which can improve the generalization of these models.

On the other hand, variational autoencoders (VAEs) [23], [24] have been proposed to capture high-variability in complex data-sets. VAEs are a class of deep-latent variable models which learn the complex intractable posterior over the data space by employing the variational inference (VI) and the reparameterization trick. However, vanilla VAEs are suited for non-sequential data-sets only. In [22], the authors extend the variational autoencoders (VAEs) for highly-variable sequential data which is named variational recurrent neural network (VRNN).

A variational recurrent neural network (VRNN) contains a variational autoencoder (VAE) at each time-step t which is conditioned on the previous hidden state $\mathbf{h}_{t-1}$ of an RNN; thus modelling the sequential structure in the data. In the same paper, the significance of this model is demonstrated on various sequential data-sets. VRNNs however, have rarely been adopted for time series forecasting tasks. Recently, in [25] the authors evaluate VRNNs for time series forecasting on various synthetic and one real benchmark data-sets against several neural baselines including recurrent neural network with extended Kalman filters (RNN-EKFs) [26], robust echo state networks (RESNs) [27] and co-evolutionary multi-task learning (CMTL) [28] and conclude that VRNNs outperform all the baselines on most data-sets.

To the best of our understanding, VRNNs have not been applied for clinical time series forecasting previously. This is particularly interesting since electronic health records (EHRs) are characterized by irregularity, sparsity and strong intricacy [12], [29]–[34]. On the other hand, electronic health records (EHRs) also provide additional domain information [5] e.g., disease diagnostics, age and ethnicity etc. beyond the primary data which can be leveraged for improved forecasting. Based on these rationales, we propose to evaluate the incorporation of multiple correlated time series in training VRNNs to achieve improved forecasting. The set of these multiple correlated time series is based on the similarity computation of the supplementary domain information, e.g., disease diagnostics and age etc. between the patients.

The rest of this paper is organized as follows. We present the basic introduction to RNN, VAE and VRNN in section II. Section III provides the blueprint to improve the VRNNs for forecasting clinical time series based on the supplementary medical information found in clinical data-sets. In section IV, we present the experimental design to empirically evaluate the effectiveness of this approach. This is followed by results in section V. Finally, we discuss the logical conclusion of the paper along-side the future research line in section VI.

II. Background

A. Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a family of neural networks which are specialized to model the temporal correlations in the data [13], [22]. In particular, a recurrent neural network (RNN) receives a variable-length input sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T)$; which it processes by computing the so called hidden state $\mathbf{h}_t$ as a function of the current input $\mathbf{x}_t$ at time t and the previous hidden state $\mathbf{h}_{t-1}$:

$$\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{h}_{t-1} \; ; \; \theta), \tag{1}$$

where f is a non-linear activation function and θ is the associated set of parameters to be optimized. The gated implementations of f result in networks known as long short-term memory (LSTM) [35] and gated recurrent unit (GRU) [36] which regulate the flow of information and prevent issues known as vanishing and exploding gradient problems [37]. RNNs model sequences by parameterizing a factorization of the joint sequence probability distribution as a product of the conditional probabilities such that:

$$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \{\mathbf{x}_i\}_{1 \leq i < t}), \tag{2}$$

and

$$p(\mathbf{x}_t \mid \{\mathbf{x}_i\}_{1 \leq i < t}) = g(\mathbf{h}_{t-1} \; ; \; \tau), \tag{3}$$

In Eqs. (2) and (3), T corresponds to the sequence length, $\{\mathbf{x}_i\}_{1 \leq i < t}$ denotes the set of inputs preceding $\mathbf{x}_t$ and g is a function mapping the hidden state $\mathbf{h}_{t-1}$ to the output conditional probability distribution parameterized by a set of parameters τ. Note that due to the space limitation, $\{\mathbf{x}_i\}_{1 \leq i < t}$ (i.e., the set of the inputs preceding $\mathbf{x}_t$) and similar notations e.g., $\{\mathbf{x}_i\}_{1 \leq i < T}$ etc. are substituted with their compact representations i.e., $\mathbf{x}_{<t}$ in the remainder of the paper.

B. Variational Autoencoder

A variational autoencoder (VAE) [23], [24] is a deep-latent variable model to approximate the complex intractable posterior over the data space. A VAE uses a set of latent variables $\mathbf{z}$ designed to capture the high-variations in the data by encoding and reconstructing the data; thereby learning the global properties of the data space. More specifically, a VAE consists of two neural networks: an inference network (a.k.a. the encoder) and a generative network (a.k.a. the decoder) respectively. The encoder encodes the input $\mathbf{x}$ to the latent variable $\mathbf{z}$, and the decoder maps this latent variable $\mathbf{z}$ back to reproduce $\mathbf{x}$. The VAE treats the conditional probability distribution $p(\mathbf{x}|\mathbf{z})$ as highly-flexible function approximation of $\mathbf{x}$. However, the mapping from $\mathbf{z}$ to $\mathbf{x}$ can't be implemented because of the intractable posterior $p(\mathbf{z}|\mathbf{x})$ on the latent variable. The VAE thus introduces the variational approximation $q(\mathbf{z}|\mathbf{x})$ of the intractable posterior $p(\mathbf{z}|\mathbf{x})$. The approximate posterior $q(\mathbf{z}|\mathbf{x})$ has highly-flexible form and its parameters are generated by the inference (i.e., encoder) network. Lastly, the variational approximation $q(\mathbf{z}|\mathbf{x})$ of $p(\mathbf{z}|\mathbf{x})$ enables the use of Evidence Lower Bound (ELBO) (a.k.a. variational lower bound) as:

$$\log p(\mathbf{x}) \geq -\mathrm{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})], \tag{4}$$

where $\mathrm{KL}(Q||P)$ is the Kullback-Leibler divergence between two probability distributions Q and P. In [23], the variational posterior $q(\mathbf{z}|\mathbf{x})$ is modelled by a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$ where the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ are the outputs of the inference network and diag corresponds to the diagonal covariance structure of the Gaussian distribution. The prior $p(\mathbf{z})$ is assumed to be a standard Gaussian distribution. The training process focuses on maximizing ELBO (4) which yields the optimal parameters for the inference and generative networks. A low variance estimator can be substituted with the help of the reparameterization trick $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon$; where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is a vector of standard Gaussian variables and $\odot$ denotes the element-wise product:

$$\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}\ |\ \mathbf{x})}[\log p(\mathbf{x}\ |\ \mathbf{z})] = \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\log p(\mathbf{x}|\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}\odot\epsilon)]. \tag{5}$$

C. Variational Recurrent Neural Network

A variational recurrent neural network (VRNN) [22] is the extension of a standard VAE discussed above to the cases with sequential data. It is a combination of an RNN and a VAE as described in Eqs. (1) and (5) respectively. More specifically, a VRNN employs a VAE at each time-step t. However, the prior on the latent variable $\mathbf{z}_t$ of this VAE is assumed to be a multivariate Gaussian whose parameters are computed from the previous hidden state $\mathbf{h}_{t-1}$ of the RNN such that:

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{0,t})), [\boldsymbol{\mu}_{0,t},\ \boldsymbol{\sigma}_{0,t}] = \varphi^{\mathrm{prior}}_{\tau}(\mathbf{h}_{t-1}), \tag{6}$$

In Eq. (6), $\boldsymbol{\mu}_{0,t}$ and $\boldsymbol{\sigma}_{0,t}$ are the parameters of the prior $p(\mathbf{z}_t)$ and $\varphi^{\mathrm{prior}}_{\tau}$ refers to a non-linear function such as a FFNN parameterized by a set of parameters τ. The generating distribution in the decoder $p(\mathbf{x}|\mathbf{z})$ is conditioned on both $\mathbf{z}_t$ and $\mathbf{h}_{t-1}$ such that:

$$\begin{aligned} \mathbf{x}_t \mid \mathbf{z}_t &\sim \mathcal{N}(\boldsymbol{\mu}_{x,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{x,t})),\\ \text{where } [\boldsymbol{\mu}_{x,t}, \boldsymbol{\sigma}_{x,t}] &= \varphi^{\mathrm{dec}}_{\tau}(\varphi^{\mathbf{z}}_{\tau}(\mathbf{z}_t), \mathbf{h}_{t-1}), \end{aligned} \tag{7}$$

In Eq. (7), $\boldsymbol{\mu}_{x,t}$ and $\boldsymbol{\sigma}_{x,t}$ are the parameters of the generating distribution. The hidden state $\mathbf{h}_t$ of the RNN is updated as:

$$\mathbf{h}_t = f(\varphi^{\mathbf{x}}_{\tau}(\mathbf{x}_t), \varphi^{\mathbf{z}}_{\tau}(\mathbf{z}_t), \mathbf{h}_{t-1}\ ;\ \theta), \tag{8}$$

where f is a non-linear activation function and $\varphi^{\mathbf{x}}_{\tau}$, $\varphi^{\mathbf{z}}_{\tau}$ and $\varphi^{\mathrm{dec}}_{\tau}$ in Eqs. (6) and (7) are the FFNNs similar to $\varphi^{\mathrm{prior}}_{\tau}$. The hidden state $\mathbf{h}_t$ is a function of both $\mathbf{x}_{\leq t}$ and $\mathbf{z}_{\leq t}$. The joint probability distribution of $\mathbf{x}$ and $\mathbf{z}$ thus becomes:

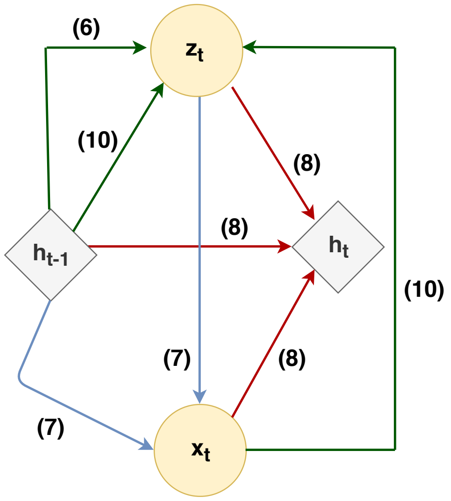

Fig. 1. The schematic view of a VRNN is presented. The green line connections correspond to the computations involving the (conditional) prior and posterior on $\mathbf{z}_t$ while the blue line connections show the computations involving the generative network, i.e., decoder. In addition, the computations for $\mathbf{h}_t$ are shown with red line connections. These connections depict the dependencies between the variables in Eqs. (6)-(10). Note that each connection/line is labelled according to the numbering of the equation it realizes.

$$p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}). \tag{9}$$

We now discuss the inference, i.e., encoder network. Here, the approximate posterior $q(\mathbf{z}_t|\mathbf{x}_t)$ is a function of both $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$ such as:

$$\begin{aligned} \mathbf{z}_t \mid \mathbf{x}_t &\sim \mathcal{N}(\boldsymbol{\mu}_{z,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{z,t})),\\ \text{where } [\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}] &= \varphi^{\mathrm{enc}}_{\tau}(\varphi^{\mathbf{x}}_{\tau}(\mathbf{x}_t), \mathbf{h}_{t-1}), \end{aligned} \tag{10}$$

where $\boldsymbol{\mu}_{z,t}$ and $\boldsymbol{\sigma}_{z,t}$ are the parameters of the approximate posterior and $\varphi^{\mathrm{enc}}_{\tau}$ is a FFNN same as $\varphi^{\mathrm{prior}}_{\tau}$, $\varphi^{\mathbf{x}}_{\tau}$, $\varphi^{\mathbf{z}}_{\tau}$ and $\varphi^{\mathrm{dec}}_{\tau}$. Conditioning on $\mathbf{h}_{t-1}$, the posterior follows the factorization:

$$q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}). \tag{11}$$

The objective function to train both, inference and generative networks is to maximize ELBO based on the factorization in Eqs. (9) and (11); giving rise to the accumulative ELBO as:

$$\begin{aligned} &\mathbb{E}_{\mathbf{z}_{\leq T}\sim q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}(-\mathrm{KL}(q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})||\\ &\quad p(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})) + \log\ p(\mathbf{x}_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}))\Big]. \end{aligned} \tag{12}$$

The graphical representation of a standard VRNN is presented in figure 1.

III. Modelling domain information in Clinical Time Series Forecasting

Clinical data-sets are characterized by loads of supplementary information accompanying the primary data [38]–[40]. Such supplementary information may contain details about the patients, the laboratory tests, and the working condition of the hospitals and ICUs [10], [41]. Some of this information may be useful for the clinical analysis, early triage, risk assessment, and a better understanding about the ongoing treatment [42]. Thus, it is critical to incorporate such supplementary information for tasks such as temporal signal forecasting, risk assessment, mortality classification for critical patients, phenotype classification [6]–[8] and length-of-stay prediction [9]. However, there is a lack of common algorithmic approaches to exploit such domain information to improve the outcome of the learning tasks.

To conduct time series forecasting for a particular patient; we propose to take a set of similar patients which is determined by some similarity criteria. Temporal signals extracted from these similar patients can be combined with the signals from the patient of interest to increase the robustness [43]–[45] of the forecasting. This can improve the generalization ability of VRNNs for two reasons. Firstly, if the input time series varies slightly; the model would be less prone to fail in reconstructing the time series by including the correlated temporal signals of the similar patients. Secondly, the model utilizing the correlated temporal signals in the learning phase would be less likely to over-fit the data. For the similarity criterion, we choose the K-Nearest Neighbours (KNNs) with respect to the cosine similarity metric on disease diagnostics.

We denote the set of correlated temporal signals for a patient at time t with $\mathbf{x}_t^{\text{rel}}$. The probability distributions for generative and inference networks are updated as:

$$p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{z}_{\leq t}, \mathbf{x}_{<t}, \mathbf{x}_{<t}^{\text{rel}}) \cdot p(\mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}^{\text{rel}}) \quad (13)$$

$$q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}) = \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{x}_{\leq t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}^{\text{rel}}). \quad (14)$$

Similarly, all the expressions in section II-C needs to be updated by additionally conditioning on multiple correlated temporal signals $\mathbf{x}_t^{\text{rel}}$. We now move on to discuss the experimental setup[‡‡] for evaluating the effectiveness of $\mathbf{x}_t^{\text{rel}}$ in the VRNN model.

IV. Experimental Setup

A. Data preprocessing

We use "Medical Information Mart for Intensive Care (MIMIC III)" [38] which is publicly available and widely accepted benchmark data-set for clinical trials. MIMIC III is a relational database containing information of approximately 60,000 ICU admissions. It contains information [10], [46] about the demographics of the patients, the laboratory tests, keynote events during the ICU stay, medications and the temporal signals in the ICU e.g., Mean Blood Pressure (MBP) and Body Temperature (BT) etc. Since MIMIC III is a highly-complicated data-set involving millions of events; it is important to follow a standard approach to preprocess the data which can be used for the learning tasks. To this end, we follow the procedure of [10] which provides the benchmark preprocessing for MIMIC III. After following [10] for preprocessing; we are left with five different data-sets extracted from MIMIC III where each data-set corresponds

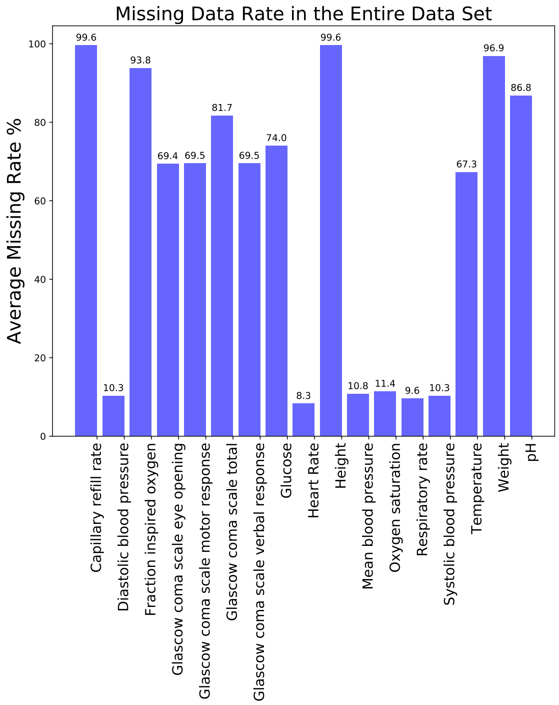

Fig. 2. Average (i.e., train and test both) missing rate % for all 17 temporal features is presented in this figure. Capillary refill rate and Height are the channels with maximum missing rate (99.6) %, while Heart Rate has lowest missing ratio (8.3) %.

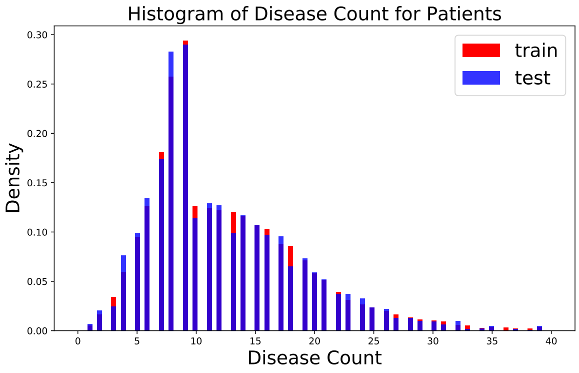

Fig. 3. The histogram of disease counts for patients in the training and test data-sets is presented in this figure. The minimum and maximum number of disease(s) for an individual patient are 1 and 39 respectively; for both train and test data-set.

[‡‡] The source code is available at: https://github.com/SibghatUllah13/VRNNs-for-Clinical-Time-Series-Forecasting

to a specific learning task in [10] such as in-hospital-mortality prediction, decompensation prediction, length-of-stay prediction, phenotype classification and multitask learning. In the following, we proceed with the in-hospital-mortality data-set extracted from MIMIC III since it filters most of the issues such as the missing ids and the length of stay. Some of the important attributes of the preprocessed in-hospital-mortality data-set are presented in Table I, in which the first four columns report the description of the data, the number of patients, the number of ICU stays and the number of observed temporal features respectively. The last two columns report the number of continuous and categorical temporal variables (i.e., features) respectively. The train and test data-sets are split in the preprocessing step with a ratio of 85% - 15%.

The in-hospital-mortality data contains the timeline of the first 48 hours of each patient's stay in the ICU. It is clear from Table I, that some patients have been admitted to the ICU more than once. We remove such duplicates from the records and make sure that each patient has exactly one ICU record. Furthermore, to handle the sporadic nature of the data; we re-sample the temporal features to have exactly one entry in one hour resulting in a total of 48 entries for each patient same as [10]. In the case there are more than one entries in an hour, we take the mean and substitute it as the only entry of the hour to make the data consistent. This results in each patient represented by a matrix of size 48×17. At this point, 83% of the entries in a patient's time series matrix are missing on average. The overall missing rate for all 17 temporal features is presented in figure 2 to further highlight the issue.

It can be observed from figure 2 that some features have extremely high missing rate and are consequently not fit for further analysis. As such, we remove them from the data and are left with only 6 temporal features, all of which are continuous with a missing rate of around 10%. After this, we also remove those patients who have more than 10% missing entries. Finally, we are left with 13400 patients in the training data-set and 2312 in the test data-set and the missing rate is reduced to 10%. The missing entries are then substituted by the column mean and thereupon we assume the complete information of each patient's time series which is a matrix of size 48×6 where the six temporal features are Diastolic Blood Pressure (DBP), Heart Rate (HR), Mean Blood Pressure (MBP), Oxygen Saturation Level (OSL), Respiratory Rate (RR) and Systolic Blood Pressure (SBP) respectively. Apart from the temporal features, we also observe the disease diagnostics of each patient. This information is later used to compute $\mathbf{x}_t^{\text{rel}}$ as discussed in the previous section. The histogram of the disease counts of all patients in the training and test data-sets is presented in figure 3.

B. Similarity Computation

MIMIC III contains a variety of supplementary information e.g., ethnicity, language, age and disease information etc. beyond the temporal features of the patients. However, most of such information is missing for the majority of the patients. Disease diagnostics is the only supplementary information present for each patient. As such, we only use the disease diagnostics as extra domain information to compute the similarity between the patients. We convert each patient's disease information into a binary vector of size 6961 where 6961 is the size of the set of all unique diseases in the entire data-set. After this, we find the set of k most similar patients for each patient based on the cosine similarity of the disease vectors. We test the values of k for $2, 3, 4,$ and 5 and find out that $k = 3$ provides the best results. Thus, all the results mentioned in the next section are achieved using $k = 3$ and $\mathbf{x}_t^{\text{rel}} \in \mathbb{R}^d$ where $d = 18$. Once we have $\mathbf{x}_t^{\text{rel}}$ available, we implement and evaluate the model.

TABLE I
THIS TABLE REPORTS SOME OF THE MOST IMPORTANT ATTRIBUTES OF THE IN-HOSPITAL-MORTALITY DATA EXTRACTED FROM MIMIC III BY FOLLOWING THE PREPROCESSING IN [10].

Type	**Patients**	**ICU stays**	**Variables**	**Cont Var**	**Cat Var**
train	15331	17903	17	13	4
test	2763	3236	17	13	4

C. Model Implementation and Evaluation

Here we consider the following variants of VRNN in our experiment:

- Vanilla VRNN,
- VRNN-I (without the conditional prior in Eq. (6)),
- The proposed approaches: VRNN-S and VRNN-I-S ("S" stands for similarity), which implement the similar data $\mathbf{x}_t^{\text{rel}}$ into VRNN and VRNN-I respectively.

We do not include the other neural baselines such as recurrent neural network with extended Kalman filters (RNN-EKFs) [26], robust echo state networks (RESNs) [27] and co-evolutionary multi-task learning (CMTL) [28] since we're fundamentally interested in robust and improved forecasting of VRNNs by attempting to learn the local variations in the data. Table II reports the implementation details of all four models. In Table II, the first three columns show the model, the dimensions of $\mathbf{x}_t$ and $\mathbf{z}_t$ respectively. The fourth and fifth column describe the number of hidden layers and the size of each hidden layer accordingly. The last two columns report the batch size and the number of epochs respectively. The implementations of all four models are with GRUs and all temporal features are re-scaled between -1 and 1. The choice of the batch size is based on [23]. For the choice of the number of hidden layers and their size, we try a variety of combinations including the previous settings in [10], [22], [23]. Our final choice of the hidden size, number of layers and number of epochs are now based on the quality of results on the test data-set as all four models performed best at the current settings reported in Table II, which is different from any of the settings in [10], [22], [23].

We are interested in evaluating our models for multi-step-ahead forecasting. We evaluate the models on one to ten-step-ahead forecasting. For one-step-ahead forecasting, we train all the models on 47 time-steps and predict the last time-step. For

TABLE II

THE IMPLEMENTATION DETAILS OF ALL FOUR MODELS ARE PRESENTED IN THIS TABLE. NOTE THAT THE IMPLEMENTATION DETAILS FOR ALL FOUR MODELS ARE THE SAME AS INDICATED BY THE NOTATION “≅” IN THE TABLE.

Model	X	Z	No. Layers	Hidden	Batch	EPOCH
VRNN	6	2	2	50	100	5
VRNN-I	≅	≅	≅	≅	≅	≅
VRNN-S	≅	≅	≅	≅	≅	≅
VRNN-I-S	≅	≅	≅	≅	≅	≅

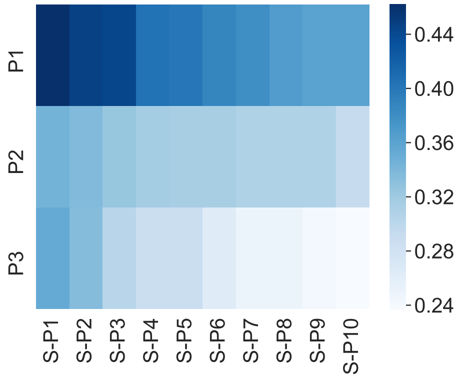

Fig. 4. This heat map visualizes the cosine similarity values between our patients of interest (P1, P2, and P3) and their corresponding ten most similar patients (S-P*) based on disease diagnostics.

two to five-step-ahead forecasting, we train all the models on 43 time-steps and predict the next two, three, four and five steps respectively. For six to ten-step-ahead forecasting, we train all the models on 38 time-steps and predict the next six, seven, eight, nine and ten steps. We evaluate all the models on Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y_i})^2} \tag{15}$$

Where y_i and $\hat{y_i}$ in Eq. (15) are the vectors representing the true and predicted values of all six temporal features for the i^{th} patient and M denotes the size of the test data-set. We now discuss the results obtained from the above experimental setup.

V. RESULTS

In this section, we first report the Average (i.e., for all the temporal variables) Root Mean Square Error (RMSE) (15) on the test data-set for multi-step-ahead forecasting in Table III. In this table, the first column displays the step size for forecasting. The next four columns present the RMSE with rounded standard deviations using VRNN, VRNN-I (i.e., without the conditional prior in Eq. (6)), VRNN-S (i.e., VRNN employing $\mathbf{x}_t^{\text{rel}}$), and VRNN-I-S (i.e., without the conditional prior and employing $\mathbf{x}_t^{\text{rel}}$). The last two columns share the p values resulting from the Mann-Whitney U test. These tests have the alternative hypotheses RMSE (VRNN-S) $<$ RMSE (VRNN) and RMSE (VRNN-I-S) $<$ RMSE (VRNN-I) respectively. These tests find if VRNNs utilizing $\mathbf{x}_t^{\text{rel}}$ (also labelled M3 and M4 in the table) are significantly better than the respective baselines (which are labelled M1 and M2 respectively in the table). From Table III, it can be observed that VRNN-I-S achieves the lowest values of RMSE in all the ten cases. Furthermore, VRNN-S achieves the second lowest error in all the ten cases. Lastly, the rounded standard deviations in Table III are analogous for all four models. From the last two columns in Table III, we find out that in 6/10 cases; at-least one of VRNN-S and VRNN-I-S performs significantly better than the respective baseline as indicated by the p values.

We further perform a simple qualitative analysis to highlight the importance of $\mathbf{x}_t^{\text{rel}}$ in robust and improved forecasting of VRNNs. We select three patients in the test data-set where VRNN-S and VRNN-I-S both achieve the lowest RMSE (15). For each of these patients, we select the ten most similar patients based on disease diagnostics and plot the corresponding cosine similarity values in the form of a heat map in figure 4. This heat map verifies that our choice of $k = 3$ in previous section is plausible since in all three cases, high similarity values are observed for the first few (i.e., two, three) related patients only. Moving forward with $k = 3$; we report the information about the set of common diseases between our selected patients and their corresponding most similar patients in Table IV. In this table, the first column shows the identity of each of the three selected patients. The second column reports the number of common diseases between that patient and its k most similar patients. The third column shares the International Classification of Diseases, Ninth Revision (ICD9) codes for the corresponding diseases. The last column categorizes the respective ICD9 codes to the most appropriate disease family (i.e, Heart, Blood Pressure, Kidney, Respiratory) for better interpretation and analysis.

After reporting the information about the common diseases, we plot the predictions of all four models on our patients of interest in figure 5. This figure shares the one-step-ahead predicted values (re-scaled) for all six temporal variables for these patients. Considering the first patient (P1) in figure 5; we observe that VRNN-S and VRNN-I-S outperform the baselines on Heart Rate (HR) which is related to the category of the most common diseases for that Patient in Table IV. Similarly analyzing the second patient (P2); we observe that VRNN-S and VRNN-I-S outperform the baselines on Systolic Blood Pressure (SBP) which is strongly related to high blood pressure related diseases. Finally, the same analysis is performed for third patient (P3) where VRNN-S and VRNN-I-S achieve superior predictions on Respiratory Rate (RR) and Systolic Blood Pressure (SBP). From figure 5, we verify that $\mathbf{x}_t^{\text{rel}}$ indeed helps improving the forecasting accuracy of the VRNNs for clinical signals. This is especially true for the temporal features which are related to the set of the common diseases between the patients. We now move on to discuss the conclusion of the paper along-side the future research line.

TABLE III

RMSE WITH ROUNDED STANDARD DEVIATIONS ON ALL TEN STEPS AHEAD FORECASTING TASKS ON TEST DATA ARE PRESENTED IN THIS TABLE. THE FIRST COLUMN SHOWS THE STEP SIZE, THE NEXT FOUR COLUMNS SHARE THE RMSE FOR ALL FOUR MODELS. GIVEN THE ALTERNATIVE HYPOTHESES H_a: M3 $<$ M1 AND H_a: M4 $<$ M2 WHERE M1, M2, M3 AND M4 CORRESPOND TO THE MODELS IN COLUMNS 2-5 RESPECTIVELY; TWO MANN-WHITNEY U TESTS ARE PERFORMED TO FIND IF THE ERROR DIFFERENCES ARE SIGNIFICANT USING STANDARD $\alpha = 0.05$ IN BOTH TESTS. THE RESULTING p-VALUES FOR BOTH STATISTICAL TESTS ARE PRESENTED IN THE LAST TWO COLUMNS.

Step Size	**VRNN (M1)**	**VRNN-I (M2)**	**VRNN-S (M3)**	**VRNN-I-S (M4)**	H_a: **M3 $<$ M1**	H_a: **M4 $<$ M2**
1	0.01152 ± 0.0034	0.01209 ± 0.0035	0.01040 ± 0.0031	**0.01034** ± 0.0030	**0**	**0**
2	0.01047 ± 0.0022	0.01047 ± 0.0022	0.01042 ± 0.0022	**0.01039** ± 0.0022	0.26	0.078
3	0.01058 ± 0.0018	0.01059 ± 0.0018	0.01053 ± 0.0018	**0.01050** ± 0.0018	0.23	**0.045**
4	0.01062 ± 0.0017	0.01062 ± 0.0016	0.01057 ± 0.0016	**0.01054** ± 0.0016	0.21	**0.036**
5	0.01062 ± 0.0015	0.01063 ± 0.0014	0.01058 ± 0.0015	**0.01055** ± 0.0015	0.22	**0.021**
6	0.01071 ± 0.0014	0.01064 ± 0.0013	0.01062 ± 0.0013	**0.01060** ± 0.0013	0.074	0.14
7	0.01071 ± 0.0013	0.01064 ± 0.0012	0.01063 ± 0.0012	**0.01060** ± 0.0012	0.056	0.12
8	0.01073 ± 0.0012	0.01066 ± 0.0011	0.01064 ± 0.0011	**0.01062** ± 0.0012	**0.046**	0.10
9	0.01074 ± 0.0012	0.01066 ± 0.0011	0.01065 ± 0.0011	**0.01062** ± 0.0011	**0.042**	0.09
10	0.01073 ± 0.0011	0.01066 ± 0.0010	0.01065 ± 0.0010	**0.01062** ± 0.0011	0.051	0.074

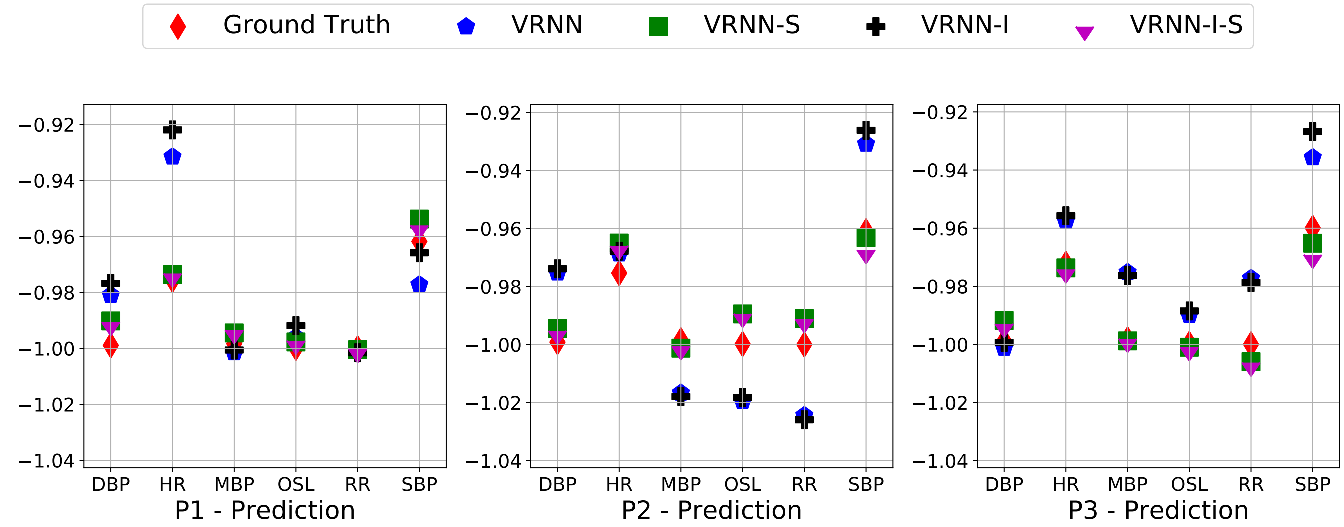

Fig. 5. One step ahead predictions on all six temporal features of the selected patients are presented in this figure. The six temporal features are Diastolic Blood Pressure (DBP), Heart Rate (HR), Mean Blood Pressure (MBP), Oxygen Saturation Level (OSL), Respiratory Rate (RR) and Systolic Blood Pressure (SBP) respectively.

TABLE IV

THIS TABLE SHARES THE INFORMATION OF THE COMMON DISEASES FOUND BETWEEN OUR SELECTED PATIENTS AND THEIR k MOST SIMILAR PATIENTS.

ID	**Dis..**	**ICD9**	**Category**
P1	4	$414(.01, .9), 427.31, 428.0$	Heart, Blood Pres..
P2	3	$785.52, 995.92, 584.9$	High Blood Pres.., Kidney
P3	2	$507.0, 518.81$	Respiratory, Blood Pres..

VI. CONCLUSIONS AND OUTLOOK

In this paper, we evaluate the effectiveness of utilizing multiple correlated time series in clinical time series forecasting tasks. Such correlated time series can be extracted from a set of similar patients; where the similarity can be computed on the basis of the supplementary domain information such as disease diagnostics, age and ethnicity etc. As our baselines, we choose VRNN and its variant which are state-of-the-art deep-generative models for sequential data-sets. From the findings in section V, we believe that the performance of VRNNs can be improved by including the correlated temporal signals. This is since in 6/10 cases considered in Table III; at-least one of VRNN-S and VRNN-I-S performs significantly better than the baselines as indicated by the p values resulting from the statistical tests. Additionally, it can be observed from figure 5 that the incorporation of multiple correlated time series helps recovering the temporal features related to the common diseases between the patients.

It it nonetheless important to state that the simple similarity criteria used in the experiments needs to be further enhanced to capture more complex relationships between the patients such as learning vector representations of graphs [47] in an unsupervised fashion. These vector representations can then be included in the training to learn more robust relationships between the patients. We aim to focus on such enhanced similarity computations and other information-rich application areas e.g., financial and economic time series forecasting etc.

in future. On the basis of the points discussed above, we believe that discarding such supplementary domain information while analyzing clinical data-sets may not be an optimal strategy since such information may be used to improve the generalization. Lastly, we believe there is a dire need of additional clinical benchmark data-sets to improve upon the state-of-the-art in this area.

Acknowledgment

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 766186. Hao Wang acknowledges the support from the Paris Île-de-France Region.

References

[1] D. W. Bates, S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar, "Big data in health care: using analytics to identify and manage high-risk and high-cost patients," *Health Affairs*, vol. 33, no. 7, pp. 1123–1131, 2014.

[2] H. C. Koh, G. Tan, *et al.*, "Data mining applications in healthcare," *Journal of healthcare information management*, vol. 19, no. 2, p. 65, 2011.

[3] H. Kaur and S. K. Wasan, "Empirical study on applications of data mining techniques in healthcare," *Journal of Computer science*, vol. 2, no. 2, pp. 194–200, 2006.

[4] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *Ieee Access*, vol. 5, pp. 8869–8879, 2017.

[5] G. D. Clifford, D. J. Scott, M. Villarroel, *et al.*, "User guide and documentation for the mimic ii database," *MIMIC-II database version*, vol. 2, no. 95, 2009.

[6] T. A. Lasko, J. C. Denny, and M. A. Levy, "Correction: Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," *PLoS one*, vol. 8, no. 8, 2013.

[7] Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu, "Deep computational phenotyping," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 507–516, ACM, 2015.

[8] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine Learning for Healthcare Conference*, pp. 301–318, 2016.

[9] 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 Medicine*, vol. 1, no. 1, p. 18, 2018.

[10] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *arXiv preprint arXiv:1703.07771*, 2017.

[11] S.-J. Bang, Y. Wang, and Y. Yang, "Phased-lstm based predictive model for longitudinal ehr data with missing values,"

[12] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[14] G. Clermont, D. C. Angus, S. M. DiRusso, M. Griffin, and W. T. Linde-Zwirble, "Predicting hospital mortality for patients in the intensive care unit: a comparison of artificial neural networks with logistic regression models," *Critical care medicine*, vol. 29, no. 2, pp. 291–296, 2001.

[15] M. Ghassemi, M. A. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits, and M. Feng, "A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[16] R. Ranganath, A. Perotte, N. Elhadad, and D. Blei, "Deep survival analysis," *arXiv preprint arXiv:1608.02158*, 2016.

[17] S. Yousefi, C. Song, N. Nauata, and L. Cooper, "Learning genomic representations to predict clinical outcomes in cancer," *arXiv preprint arXiv:1609.08663*, 2016.

[18] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[19] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *Advances in Neural Information Processing Systems*, pp. 7785–7794, 2018.

[20] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," *arXiv preprint arXiv:1411.7610*, 2014.

[21] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," *arXiv preprint arXiv:1206.6392*, 2012.

[22] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, pp. 2980–2988, 2015.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[24] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[25] D. Hsu, "Multi-period time series modeling with sparsity via bayesian variational inference," *arXiv preprint arXiv:1707.00666*, 2017.

[26] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks," *IEEE Transactions on neural networks*, vol. 5, no. 2, pp. 279–297, 1994.

[27] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust echo state network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 787–799, 2012.

[28] R. Chandra, Y.-S. Ong, and C.-K. Goh, "Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction," *Neurocomputing*, vol. 243, pp. 21–34, 2017.

[29] C. Preda, A. Duhamel, M. Picavet, and T. Kechadi, "Tools for statistical analysis with missing data: application to a large medical database," *Studies in health technology and informatics*, vol. 116, p. 181, 2005.

[30] L. Marston, J. R. Carpenter, K. R. Walters, R. W. Morris, I. Nazareth, and I. Petersen, "Issues in multiple imputation of missing data for large general practice clinical databases," *Pharmacoepidemiology and drug safety*, vol. 19, no. 6, pp. 618–626, 2010.

[31] F. Cismondi, A. S. Fialho, S. M. Vieira, S. R. Reti, J. M. Sousa, and S. N. Finkelstein, "Missing data in medical databases: Impute, delete or classify?," *Artificial intelligence in medicine*, vol. 58, no. 1, pp. 63–72, 2013.

[32] K. J. Janssen, A. R. T. Donders, F. E. Harrell Jr, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons, "Missing covariate data in medical research: to impute is better than to ignore," *Journal of clinical epidemiology*, vol. 63, no. 7, pp. 721–727, 2010.

[33] J. Dauwels, L. Garg, A. Earnest, and L. K. Pang, "Tensor factorization for missing data imputation in medical questionnaires," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2109–2112, IEEE, 2012.

[34] J. A. Sterne, I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter, "Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls," *Bmj*, vol. 338, p. b2393, 2009.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[37] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, pp. 1310–1318, 2013.

[38] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[39] S. Shah, D. Ledbetter, M. Aczon, A. Flynn, and S. Rubin, "2: Early prediction of patient deterioration using machine learning techniques with time series data," *Critical Care Medicine*, vol. 44, no. 12, p. 87, 2016.

[40] C. S. Carlin, L. V. Ho, D. R. Ledbetter, M. D. Aczon, and R. C. Wetzel, "Predicting individual physiologically acceptable states at discharge from a pediatric intensive care unit," *Journal of the American Medical Informatics Association*, vol. 25, no. 12, pp. 1600–1607, 2018.

[41] M. Aczon, D. Ledbetter, L. Ho, A. Gunny, A. Flynn, J. Williams, and R. Wetzel, “Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks,” *arXiv preprint arXiv:1701.06675*, 2017.

[42] Y. Liu, B. Logan, N. Liu, Z. Xu, J. Tang, and Y. Wang, “Deep reinforcement learning for dynamic treatment regimes on medical registry data,” in *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 380–385, IEEE, 2017.

[43] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*, vol. 28. Princeton University Press, 2009.

[44] H.-G. Beyer and B. Sendhoff, “Robust optimization–a comprehensive survey,” *Computer methods in applied mechanics and engineering*, vol. 196, no. 33-34, pp. 3190–3218, 2007.

[45] S. Ullah, H. Wang, S. Menzel, B. Sendhoff, and T. Back, “An empirical comparison of meta-modeling techniques for robust design optimization,” in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 819–828, IEEE, 2019.

[46] T. Gentimis, A. Ala’J, A. Durante, K. Cook, and R. Steele, “Predicting hospital length of stay using neural networks on mimic iii data,” in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pp. 1194–1201, IEEE, 2017.

[47] A. G. Duran and M. Niepert, “Learning graph representations with embedding propagation,” in *Advances in neural information processing systems*, pp. 5119–5130, 2017.