Data Pre-Processing Using Neural Processes for Modeling Personalized Vital-Sign Time-Series Data

Pulkit Sharma, Farah E. Shamout, Vinayak Abrol, and David A. Clifton

***Abstract*—Clinical time-series data retrieved from electronic medical records are widely used to build predictive models of adverse events to support resource management. Such data is often sparse and irregularly-sampled, which makes it challenging to use many common machine learning methods. Missing values may be interpolated by carrying the last value forward, or through linear regression. Gaussian process (GP) regression is also used for performing imputation, and often re-sampling of time-series at regular intervals. The use of GPs can require extensive, and likely adhoc, investigation to determine model structure, such as an appropriate covariance function. This can be challenging for multivariate real-world clinical data, in which time-series variables exhibit different dynamics to one another. In this work, we construct generative models to estimate missing values in clinical time-series data using a neural latent variable model, known as a Neural Process (NP). The NP model employs a conditional prior distribution in the latent space to learn global uncertainty in the data by modelling variations at a local level. In contrast to conventional generative modelling, this prior is not fixed and is itself learned during the training process. Thus, NP model provides the flexibility to adapt to the dynamics of the available clinical data. We propose a variant of the NP framework for efficient modelling of the mutual information between the latent and input spaces, ensuring meaningful learned priors. Experiments using the MIMIC III dataset demonstrate the effectiveness of the proposed approach as compared to conventional methods.**

***Index Terms*—Neural processes, Gaussian processes, data interpolation, medical data, deep learning.**

Manuscript received August 13, 2020; revised May 3, 2021, July 25, 2021, and August 15, 2021; accepted August 16, 2021. Date of publication August 30, 2021; date of current version April 13, 2022. This work was supported by the National Institute for Health Research (NIHR) and Oxford Biomedical Research Centre (BRC). *(Pulkit Sharma and Farah E. Shamout contributed equally to this work.) (Corresponding author: Pulkit Sharma.)*

Pulkit Sharma is with the Institute of Biomedical Engineering, Department of Engineering Science, University of Oxford, OX3 7DQ Oxford, U.K. (e-mail: pulkit.sharma@eng.ox.ac.uk).

David A. Clifton is with the Department of Engineering Science, University of Oxford, OX3 7DQ Oxford, U.K., and also with the Oxford-Suzhou Centre for Advanced Research, Suzhou 215123, China (e-mail: david.clifton@eng.ox.ac.uk).

Farah E. Shamout is with the Engineering Division, New York University Abu Dhabi, Abu Dhabi 129188, UAE (e-mail: fs999@nyu.edu).

Vinayak Abrol is with the Mathematical Institute, Andrew Wiles Building, OX2 6GG Oxford, U.K. (e-mail: vinayak.abrol@maths.ox.ac.uk).

Digital Object Identifier 10.1109/JBHI.2021.3107518

I. Introduction

THE widespread use of electronic health record (EHR) systems has resulted in an increased acquisition of digital clinical time-series data [1]. For each patient, various types of data are routinely recorded during a hospital encounter and stored in the EHR. This includes vital signs, lab test results, diagnoses, and medications administered; resulting in sequences of clinical observations describe a patient's health trajectory. Such data present an opportunity to use machine learning methods to support clinical decision-making, such as via risk assessment models [2], [3] or predicting the patient's length of stay [4]–[6].

The measurement frequency of data varies significantly across patients, between different clinical variables, and over time. These observations are typically represented as a sequence of discrete, fixed-width time steps resulting in sequences with missing values, where substantial sparsity is common in real-world healthcare data [7]. In the past, various approaches have been developed to address this issue in the clinical time series data [4], [8]–[11]. One solution could be to remove the missing data an use the observed data for analysis. This could result in performance degradation, specially when missing rate is high and inadequate samples are kept. Often, a data imputation method is employed to substitute the missing values in the clinical time series. Another solution is to fill in the missing values with substituted values, which is known as data imputation. Most existing works employ heuristics or unsupervised interpolation techniques to estimate the missing values. Linear interpolation or carrying the most recent value forward are generally applied to deal with missing entries in data [4], [8], [12]–[14]. The most common reason to use this method in clinical research is that it conforms to the data collection methodology, e.g., generally the measurements are taken by nurses when there is a change in clinical measurements.

This approach results in unbiased estimates when the data is missing complete at random and can still provide conservative estimates in some conditions. For example, if the condition of a patient is stable, there may not be many changes in clinical measurements (recorded by nurses). Hence, there may not be many clinical measurements recorded after an initial observation where the measurements seems to be stable. Thus, this recorded clinical observation can be carry forward till the next recorded observation. Let us consider another case, when the patient is expected to deteriorate in the future, in general, one expect some

2168-2194 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

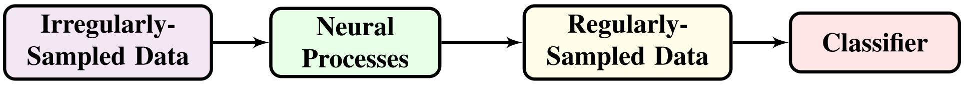


Fig. 1. Block diagram of training in the proposed framework. NPs are employed on irregularly-sampled data to obtain regularly-sampled data, which are presented to a classifier.

changes in the vital signs. Since the measurements are taken (by nurses) when there is a change in clinical variable, we are likely to have an observation when that change occurs. Now, if the measurement stays around the change in future too, it is likely that there are no recorded observations and again one can carry forward this to the next recorded observation. Although the carry forward method of imputation, conforms to data collection methodology it does not account for the uncertainty in the interpolated data and may result in bias and error [15]. To address this non-parametric probabilistic methods such as Gaussian process regression has proven to be a robust choice to model physiological data [8]–[11]. One advantage of GPs is that they can be optimized exactly by fine-tuning the hyperparameters (such as the bandwidth of a Gaussian kernel), and this often allows a fine and precise trade-off between fitting the data and capturing the uncertainty. Although GPs can be well-tuned without much prior knowledge, computation time for GP regression scales cubically in the number of data points. Further, to capture a complicated non-smooth function, one needs an approach that can scale to large datasets and generalize non-locally. Recent advancements in deep learning with application-specific architectures are more attractive with respect to these two properties.

In this work, we propose using neural processes, a neural network-based probabilistic model which can represent a distribution over stochastic processes, to model physiological time-series data [16], [17]. To illustrate the applicability of the NP model, we use it to extract regularly-sampled[1] observations for each of the vital signs, which are then used as input features to common benchmark tasks, namely predicting in-hospital mortality, identifying deterioration, or classifying phenotype [4]. The block diagram depicting the proposed framework is described in Fig. 1. As illustrated, the framework consists of two main components: (i) a generative model to derive the regularly-spaced samples for each of the vital signs and (ii) a classifier to compute the probability of an adverse event. Similar to GPs [18], NPs are probabilistic in nature as the model learns a distribution over a wide class of non-linear functions and captures uncertainty in its predictions. The main difference between NPs and GPs is that an NP learns a data-dependent prior rather than the user-defined prior associated with the mean and covariance functions of the GP [16]. For any task, using a GP requires an additional optimization procedure to identify the most suitable kernel and its corresponding hyperparameters. In contrast, NPs are trained in an end-to-end manner and are completely data-driven, which is appealing when handling complex multi variate time-series data. Further, for effective modelling using a NP, we proposed to use a modified objective function to render it suitable for medical time-series modelling (see Section III). We performed extensive experiments using a publicly-available dataset to show that the proposed NP-based generative modeling framework performs comparable to existing GP-based methods, while offering the advantage of data-driven discovery of model structure, as introduced above.

The main contributions of the paper are:

- Development and validation of an NP-based model for clinical time-series data by learning a data-driven prior.
- Adopting a personalised training strategy through pre-training a population-based model and fine-tuning for individual patient data.
- Incorporation of Maximum-Mean Discrepancy (MMD), in contrast to the conventional Kullback-Leibler divergence objective function, during NP model training to maximize information sharing between the latent and prior distributions.
- Application of NP-based processing to common clinical benchmark predictive tasks using a variety of deep learning classification methods on a public dataset (MIMIC III) to allow for reproducibility.

The remainder of the paper is organized as follows: Section II briefly reviews GPs and NPs; Section III describes the proposed approach of employing an NP to derive regularly-sampled vital-sign time-series; the experimental setup is discussed in Section IV; experimental results are discussed in Section V; and a conclusion is presented in Section VI.

II. Background

Consider a training data set O of a univariate time-series variable consisting of $\{(x_i, y_i)\}_{i=1}^{N}, \subset X \times Y$, with $\mathbf{x} = \{x_i\}_{i=1}^{N} \subset X$ as input and $\mathbf{y} = \{y_i\}_{i=1}^{N} \subset Y$ as output. Let us assume functions $f : X \to Y$, such that $y_i = f(x_i)$, where $f(.)$ is the unseen true underlying function. The goal of inference with a GP [19] or an NP [16] is to estimate the posterior distribution over $f(.)$ and employ it to estimate predictive densities at another set of $\mathbf{x}^* = \{x_j\}_{j=1}^{J} \subset X$ unlabelled points, i.e., $f(x_j)$.

A. Brief Review of Gaussian Processes

This section provides a brief overview of GP regression, while more details can be found in [19]. GP regression, a widely accepted data-modelling technique in health informatics [9], offers a probabilistic approach to modelling time-series data. We assume that the underlying true function $f(.)$ has a Gaussian prior:

$$f(x) \sim \mathcal{N}(m(x), k(x, x')) \tag{1}$$

[1] Regularly-sampled here means that the all the observations are collected across uniform time scale. For example, clinical variables like heart rate, temperature etc. all are available at a particular time instance and are also estimated uniformly across the time.

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

where $m(x)$ is the mean function and $k(x, x')$ is the covariance function that describes the relationship between the observed values at x and x' using the input values (i.e., x and x'). One most commonly adopts the radial basis function (RBF) with additive noise as the covariance function, as it has been shown to work well with routinely-collected vital-sign data [20], defined as:

$$k(x, x') = \sigma_r^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) + \sigma_n^2 \delta(x, x') \qquad (2)$$

where $\delta(x, x')$ is the Kronecker delta function, l is the length scale, σ_r is the variance of the RBF, and σ_n is the variance of the additive Gaussian noise. We use Θ to denote the set of GP hyperparameters. The values of the hyperparameters are estimated by minimizing the negative log marginal likelihood using the training data $\mathbf{x}$ and $\mathbf{y}$:

$$\log\left[p(\mathbf{y}|\mathbf{x}, \Theta)\right] = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi) \qquad (3)$$

where K represents the similarity measure between pairs of training values. Since $\mathbf{y}$ and $\mathbf{y}^*$ are assumed to have a joint probability distribution [19], we obtain the posterior $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{x}, \mathbf{y})$, where $\mathbf{x}^*$ is the set of unlabelled data points, such that:

$$\mathbf{y}^* = K_* K^{-1}\mathbf{y} \qquad (4)$$

with variance:

$$\boldsymbol{\sigma} = K_{**} - K_* K^{-1} K_*^T \qquad (5)$$

where K_* represents the covariates evaluated at all pairs of the training and missing points, and K_{**} represents the covariates evaluated at all pairs of missing points.

B. Brief Review of Neural Processes

An NP is considered to be a parametric stochastic process that defines distributions over $f(.)$ for given inputs [16], [17]. The training procedure for NP involves splitting the input data O into two sets: a context set $\{(x_c, y_c)\}_{c=1}^C$ and a target set $\{(x_t, y_t)\}_{t=1}^T$ such that $C, T \subset N$, $C \cup T = N$, and $C \cap T = \Phi$, where Φ is an empty set. The NP model is then presented with C context points to estimate the corresponding function values for the T target points; i.e., $y_t = f(x_t)$. In other words, in an NP, the target set is conditional on the context set, as will be detailed below. A model can accurately predict across the entire dataset if it can learn a distribution that spans all of the underlying functions assumed to generate the training data.

At test time, an NP takes into account the context set via a finite-dimensional embedding of mappings from x to y, known as the latent space l. This latent space l is a random variable and allows the NP to capture uncertainty over functions resulting in a generative Bayesian model. The posterior distribution of l from the trained model is used as a prior to make predictions during the test time.

In detail, the NP uses the following architecture [16]:

- **Encoder** e takes in pairs of (x, y) as input and produces a representation $h' = e(x, y)$; $e : X \times Y \to \mathbb{R}^m$ is parameterised as a neural network.
- **Aggregator** a obtains an order-invariant global representation using a mean function as $h = a(h_i^i) = \frac{1}{N}\sum_i^N h_i'$. h is further used to parameterise the latent distribution $l \sim \boldsymbol{N}(\mu(h),\ I\sigma(h)), l \in \mathbb{R}^r$.
- **Decoder** d takes target locations x_t and the sampled global latent variable l as input and outputs the predictions $\hat{y}_t$ for the corresponding values of $f(x_t) = y_t$, $d : X \times \mathbb{R}^r \to \mathbb{R}^k$ which is parameterised as a neural network.

The NP uses the encoder e followed by the aggregator to estimate a a fixed representation $h_c \in \mathbb{R}^m$ from context points (x_c, y_c). h_c is used to parameterise the latent distribution. The next step involves concatenating a sample from the latent distribution with target set $\{x_t\}$ resulting in the final latent embedding x_l. The decoder network d use this embedding as input to obtain a sample from the predictive distribution of output of the target set $\hat{y}_t$. However, the latent distribution of the context set might not match the underlying distribution of training data. Thus, a latent distribution is also estimated using all the training data $\{(x_i, y_i)\}_{i=1}^N$, and a similarity metric is used among the two distributions of context set and total training data in the loss function [16]. These steps are iterated using a random subset as context in different iterations.

The training of an NP is achieved by jointly optimizing the overall loss function:

$$\mathcal{L}_{NP} = \mathcal{L}_E + \alpha\mathcal{L}_{KL} \qquad (6)$$

where $\mathcal{L}_E$ is the data error loss (or the expected log-likelihood over the target set) as defined below for optimizing the neural networks e and d, $\mathcal{L}_{KL}$ is a regularizing loss, and α is a scaling constant. The loss term $\mathcal{L}_E$ can be computed using the cross-entropy between the original (y_t) and the predicted $(\hat{y}_t)$ output, given the latent representation x_l, i.e.,

$$\mathcal{L}_E = \sum_{t=1}^{T} \log p(y_t|x_l) \qquad (7)$$

The regularizing loss $\mathcal{L}_{KL}$ aims to minimize the loss between the true prior latent distribution of all data $p(l \mid x_{1:N}, y_{1:N})$ and the latent distribution of the context set $p(l \mid x_{1:C}, y_{1:C})$. Since the computation of these distributions is intractable, an approximation is undertaken by using corresponding variational posteriors $q(l \mid x_{1:N}, y_{1:N})$ and $q(l \mid x_{1:C}, y_{1:C})$, respectively. The regularizing loss is defined as:

$$\mathcal{L}_{KL} = \log\frac{q(l \mid x_{1:C}, y_{1:C})}{q(l \mid x_{1:N}, y_{1:N})} \qquad (8)$$

where the posteriors are parameterised as $q(l \mid \cdot) = \boldsymbol{N}(\mu_l, \sigma_l)$. The regularizer $\mathcal{L}_{KL}$ is the negative Kullback Leibler (KL) divergence between $q(l \mid x_{1:N}, y_{1:N})$ and $q(l \mid x_{1:C}, y_{1:C})$. This formulation of KL divergence uses an approximate conditional prior $q(l \mid x_{1:C}, y_{1:C})$ rather than a fixed standard Gaussian prior, and is also learned during the overall training process. Employing both $\mathcal{L}_E$ and $\mathcal{L}_{KL}$ together, the overall objective function of the NP becomes:

$$\mathbb{E}_{q(l|x_{1:N}, y_{1:N})}\left[\sum_{t=1}^{T}\log p(y_t \mid l, x_t) + \log\frac{q(l \mid x_{1:C}, y_{1:C})}{q(l \mid x_{1:N}, y_{1:N})}\right] \qquad (9)$$

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

III. Neural Processes for Medical Time-Series Prediction

In this work, we show that an NP can be used for function approximation, and then to estimate missing values and resample to a constant time grid, such that regular ML classifiers may then be used. The vital signs for different individuals can vary significantly depending upon various factors such as age or the presence of various medical conditions. This makes the use of a NP more suitable than a conventional GP, as the training data are used to derive a data-dependent prior for the NP, without the need to define a fixed prior. In addition, researchers using sequence modeling methods, such as basic Long Short Term Memory (LSTM) models, generally use data points that are regularly-sampled across time.[2] Similar to the case with a GP, this issue can also be addressed using an NP model by deriving regularly-sampled data from the original irregularly-sampled clinical data.

The NP model is similar to conventional variational models such as autoencoders and has several limitations [21]. First, the approximate latent distribution is often significantly different from the true distribution. It has been shown that the objective function in eq. (6) tends to favour fitting the data distribution (due to the KL divergence term) over performing correct amortized inference between latent distributions [21], [22]. Another problem is that of the issue of less informative latent representations [23], in which the latent space may not be a meaningful representation of the original space.

In order to address these issues, we extend NP inference with an MMD-based regularizer [24], [25], instead of the canonically-employed KL divergence, to perform better matching between the latent and prior distributions. MMD compares the moments from two distributions to quantify the similarity between them. It is motivated by the notion that distances between distributions can be represented as distances between mean embeddings from these distributions. In addition, MMD has been shown to improve the discriminative power of latent representations as it encourages disentanglement [21]. It also acts as a pseudo-measure that maximizes the mutual information between the input and latent space. The use of MMD loss is popular in various works such as variational lossy autoencoders [21], [23], [26], information maximizing GANs [22], and representation learning [27].

In its general form MMD is defined as [24]

$$M(p\|q) = \sup_{f\in F}(\mathbb{E}_p[f(a)] - \mathbb{E}_q[f(b)]), \tag{10}$$

where divergence $M(p\|q)=0$ if $p=q$, only when $F=\{f, \|f\|_{\mathcal{H}} \leq 1\}$ is a unit ball in a Reproducing Kernel Hilbert Space $\mathcal{H}$ [28], [29]. Hence, given samples $a_1 \ldots a_S \sim p$ and $b_1 \ldots b_T \sim q$, and with a positive semi-definite kernel $\mathcal{K}(\cdot,\cdot)$, under the unit ball assumptions on the evaluation function, we have

Algorithm 1: NP-Based Data Interpolation for Medical Time-Series Data.

```
Input: $\mathcal{X} = [(x_i, y_i)_k,\ i = 1,\ldots,N; k = 1,\ldots,K]$
Output: $\hat{y}_j,\ \forall\ x_j, j = 1,\ldots J$.
  NP training
  For each epoch:
    For each patient ($k$)
      Randomly select set $C$ and $T$
      $(x_c, y_c),\ c \in C$;
      $x_t,\ t \in T$
      $h_c \leftarrow a(e(x_c, y_c)), h_n \leftarrow a(e(x_i, y_i))$
      $l_c \leftarrow \mathcal{N}(\mu(h_c), I\sigma(h_c)), l_n \leftarrow \mathcal{N}(\mu(h_n), I\sigma(h_n))$
      $\hat{y}_t \leftarrow d(x_t, l_c)$
      Compute Loss in (9) and Update a() and d()
  Data interpolation
  $l_c \leftarrow \mathcal{N}(\mu(h_c), I\sigma(h_c))$
  $\hat{y}_j \leftarrow d(x_j, l_c)$
```

$$\begin{aligned} M(p\|q) &= \|\mu_p - \mu_q\|^2_{\mathcal{H}} \\ &= \mathbb{E}_{p(a),p(b)}[\mathcal{K}(a,b)] - 2\,\mathbb{E}_{p(a),q(b)}[\mathcal{K}(a,b)] \\ &\quad + \mathbb{E}_{q(a),q(b)}[\mathcal{K}(a,b)] \\ &= \frac{1}{S^2}\sum_{i}^{S}\sum_{i'}^{S}\mathcal{K}(a_i, a_{i'}) - \frac{2}{ST}\sum_{i}^{S}\sum_{j}^{T}\mathcal{K}(a_i, b_j) \\ &\quad + \frac{1}{T^2}\sum_{j}^{T}\sum_{j'}^{T}\mathcal{K}(b_j, b_{j'}). \end{aligned} \tag{11}$$

The training of our proposed NP model is achieved by jointly optimizing the loss function:

$$\mathcal{L}_{NP} = \mathcal{L}_E + \alpha\mathcal{L}_M, \tag{12}$$

where $\mathcal{L}_E$ as before is the cross-entropy loss for optimizing the data error loss in the encoder (e) and decoder (d) networks, $\mathcal{L}_M$ is the MMD loss, and α is a scaling constant. We have considered the Gaussian kernel of width σ for computing the MMD loss via the kernel trick, such that

$$\mathcal{K}(a,b) = \exp\left(\frac{-\|a-b\|^2}{2\sigma}\right) \tag{13}$$

However, other handcrafted positive semi-definite kernels satisfying the Mercer's theorem can also be employed, which we defer to future work.

The overall model training uses the same strategy as the standard NP model where the input data are split into two sets: context set and target set. Algorithm 1 shows the proposed algorithm for missing-value estimation in medical time-series data using NP. The input of the algorithm is a time-series data $(x_i, y_i),\ i = 1,\ldots,N$, where x_i corresponds to the time axis and y_i is the respective time-series value. In addition, there are certain time instances $x_j,\ j = 1,\ldots,J$ with missing data. The

[2] Theoretically, sequence modeling methods can be used with irregularly sampled data, however in practice, researchers prefer to use regularly sampled data.

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

goal of this algorithm is to predict y_j for respective x_j, which is achieved in two steps. The first step involves training a NP on the available data (x_i, y_i); here the data are randomly split into context (x_c, y_c) and target set (x_t, y_t). An encoder network e is employed on the context set and the total data to derive representations which are further aggregated using an aggregator a to derive h_c and h_n, respectively. The aggregated representation h_c is used to derive a Gaussian distribution that is used to sample[3] a representation l_c. The representation l_c is concatenated to the target set x_t and fed to a decoder network d to estimate the respective output prediction values $\hat{y}_t$ (only for the target set). The same process is repeated for a number of epochs to train the NP. The second step of this algorithm involves estimating the missing values of the data at time instances $x_j,\ j = 1, \ldots, J$. This step involves sampling a representation l_c from the Gaussian distribution obtained using $\mathcal{N}(\mu(h_c),\ I\sigma(h_c))$. The representation l_c is concatenated with the time instance x_j and passed through the learned network d to estimate the respective data value y_j.

IV. Experimental Setup

This section provides the description of the dataset and different tasks used in the experimental study. The specification of GP- and NP-based models for data interpolation are also discussed. The description of different classifiers used in this paper is also provided here.

A. Dataset

We evaluated the proposed framework using the MIMIC-III database, which is the largest publicly-available clinical data in an ICU setting [30]. The widespread use of the dataset by a large community of researchers makes MIMIC-III a suitable candidate to benchmark our method. MIMIC-III database containing more than 31 million clinical events covering 42,276 ICU stays of 33,798 unique patients. Each patient's data has been divided into separate episodes containing both time-series of events, and episode-level outcomes [4]. MIMIC-III dataset have 17 clinical variables: heart rate (HR), systolic blood pressure (SBP), diastolic blood pressure (DBP), mean blood pressure (MBP), respiratory rate (RR), temperature (TEMP), and oxygen saturation (SpO_2), capillary refill rate (CRR), fraction inspired oxygen (FIO), Glascow coma scale eye opening (G-CSEO), Glascow coma scale motor response (G-CSMR), Glascow coma scale total (G-CST), Glascow coma scale verbal response (G-CSVR), Glucose, pH, weight and height. Let us assume that $\mathcal{U}$ represents these 17 variables in MIMIC III and $\mathcal{V} \subset \mathcal{U}$ represents the subset of seven vital signs: HR, SBP, DBP, MBP, RR, TEMP, and SpO_2. This study considers the subset $\mathcal{V}$ of vital signs for NP-based proposed data interpolation. The rest of the 10 variables (in subset $\mathcal{U} - \mathcal{V}$) in the MIMIC-III dataset are highly sparse with a high degree of missingness, which prohibits modeling them meaningfully using GP/NP approaches. In our experiments, we observed that when the sparsity of the features is greater than 77%, carry forward work better than the proposed approaches. Thus, for predictive modeling tasks, one may use carry-forward for highly sparse variables where the proposed method could not work effectively.

However, after interpolating the data, the regularly-sampled data in $\mathcal{V}$ is used also in downstream classification tasks (like in-hospital mortality prediction). For these, downstream tasks we also used the 10 variables in $\mathcal{U} - \mathcal{V}$ which are not regularly-sampled using the proposed NP based method. Hence, the 10 variables in $\mathcal{U} - \mathcal{V}$ are re-sampled as suggested by Harutyunyan *et al.* [4]. Missing values in features $\mathcal{U} - \mathcal{V}$ are interpolated using the previous values if they existed, otherwise we used the pre-specified normal values specified in [4]. Categorical variables were encoded using a one-hot vector.

The train-test split in our experiments is 85%-15% with patient level partitioning. We would like to clarify that in order to avoid any potential data leakage, the proposed NP-based imputation method uses the same train-test split as in the classification tasks. Thus, the NP model is trained using the train set for the classification task under consideration. The results are computed with a 95% confidence interval, which is obtained by resampling the test set K times with replacement. We set K = 1,000 for decompensation, and K = 10,000 for in-hospital mortality and phenotype prediction.

The number of time-stamped observations vary per patient episode. Therefore, we train a single population-based NP model using training data separately for each vital sign, considering time-series with at least 40 recorded time instances. The number of recorded time instances here (40) is obtained empirically. One possible reason could that, the patients with more recorded observations are generally more volatile, while patients with fewer observations are generally less volatile. Thus, it is relatively easy to model the variations when there is more data with more recorded variations. For training, a random $70 - 30\%$ split for context and target set per example is used in each iteration of model update, and for a test example all the available data points are used as context set. The input to a NP-based model is irregularly-sampled and the output is a regularly-sampled time-series with an observation every six minutes, resulting in 10 time-steps per hour. Finally, the population-based model trained on training data is fine-tuned to a single patient-specific model. This is done in order to efficiently model the patient specific dynamics. This strategy can be used to deploy the model as retraining the personalised model in not expensive and can be done in real time. Although the deployment could be problematic if we do not have any data to train the population based model. In this case, one could collect some training data before building a population based model. Alternatively, one could use an online learning to update the model parameters if there is a continuous stream of data.

One could model the data as multivariate time series in NP, but since the data considered have varying sampling rates it is difficult to model all the dependencies. Building models which can capture multiple temporal dependencies directly from multivariate-time series data is still an open problem. We found

[3] Samples of the derived Gaussian distribution from h_c and h_n are used to match the two distributions using MMD.

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

the same during our initial attempts with NP model and found univariate modeling to be working the best.

B. Task Description

The time-series data are first modelled using the NP, and then used as regularly-sampled input for several classification tasks to demonstrate its utility. The tasks considered here are in-hospital mortality, decompensation prediction, and phenotype classification, as in previous studies [4] and described below:

- **In-hospital mortality:** In acute care settings, mortality is a primary event of interest and early detection of at-risk patients is of critical importance to improving patient care. The task of mortality prediction is formulated as a binary classification problem, where the target label corresponds to patient's death before hospital discharge. The sample size for this task is 21,139 with around 13.23% mortality cases and first 48-hour of data is used for modeling. All ICU stays where length of stay is less than 48 hours or is unknown are not used.
- **Decompensation prediction:** This involves estimating whether there will be a rapid deterioration in the patient's health, within the next 24 hours. Following work in [4], [31], decompensation is formulated as binary classification with mortality in the next 24 hours at the current timestamp as target label. These decompensation labels are assigned hourly, starting at four hours after the ICU admission and ending at patient discharge or mortality. This result in a sample size of 3,431,622 with around 2.06% decomposition cases for this task.
- **Phenotyping classification:** This involves estimating which of 25 acute care (including 12 critical) conditions are present in an ICU stay record [4], [32]. The diseases can co-occur together and hence phenotyping is formulated as a multi-label classification problem. The phenotype classification is done using the full ICU stay data. This is primarily due to the unavailability of timestamps about the disease diagnosis in MIMIC-III. Thus, it is uncertain when the patient first became symptomatic or was diagnosed or first became symptomatic. This methodology is in line with the benchmarking done in literature [4] and the sample size here is 41,902.

C. GP and NP Model Specification

As a baseline to compare the curve fitting performance of NP we consider an equivalent GP method. For the GP experiments, we adopted lognormal distributions as priors to constrain each hyperparameter to be clinically meaningful [33]. The lognormal distributions chosen as priors for l were $(\mu = 1.5, \sigma = 0.1)$ for HR, RR, TEMP, and SpO_2, and $(\mu = 1.0, \sigma = 0.1)$ for SBP. The lognormal distributions chosen as priors for σ_r were $(\mu = 3.5, \sigma = 0.1)$ for HR, SBP, and SpO_2, $(\mu = 1.5, \sigma = 0.1)$ for RR, and $(\mu = 0.5, \sigma = 0.1)$ for TEMP. The lognormal distributions chosen as priors for σ_n were $(\mu = 0.0, \sigma = 0.1)$ for RR, $(\mu = 0, \sigma = 20)$ for TEMP, $(\mu = 1.5, \sigma = 0.1)$ for HR, SBP, and SpO_2.

The NP architecture consisted of an encoder network with three fully-connected layers having 150, 100 and 200 nodes with ReLU activation [34]. Similarly, the decoder of the NP model is also a three-layer network with 64, 128 and 64 nodes with sigmoid activation. The dimensions of h and l are m=80 and r=100 respectively and 50% of the available time instances are randomly selected to be context points at each epoch. All of the hyperparameters for the proposed NP model were estimated empirically, and an early-stopping method was used if the mean and variance of the mean squared error on the target set were less than 0.4 and 0.001, respectively. Some of the other experiment-specific settings are described in respective sections.

D. Classification Models

The three tasks considered in this study involve classification and to this aim we compare CNN, LSTM and CNN-LSTM classifiers [35]. The trained NP/GP model is first used for data interpolation using the vital signs in subset $\mathcal{V}$, followed by normalization into the range [-1,1] before being presented to the classifier.

1) Classifier 1:

The architecture of the LSTM classifier differed for individual tasks for a fair comparison with existing methods in [4]. In particular, for mortality prediction the model consisted of two layers; a bi-directional layer with 8 nodes followed by a LSTM layer with 16 nodes. For the decompensation and phenotyping tasks, the model consisted of one LSTM layer with 128 and 256 nodes, respectively. In each model, LSTM layers were trained using a dropout of 0.3. The final layer in each model is a fully-connected dense layer with one node for classification.

2) Classifier 2:

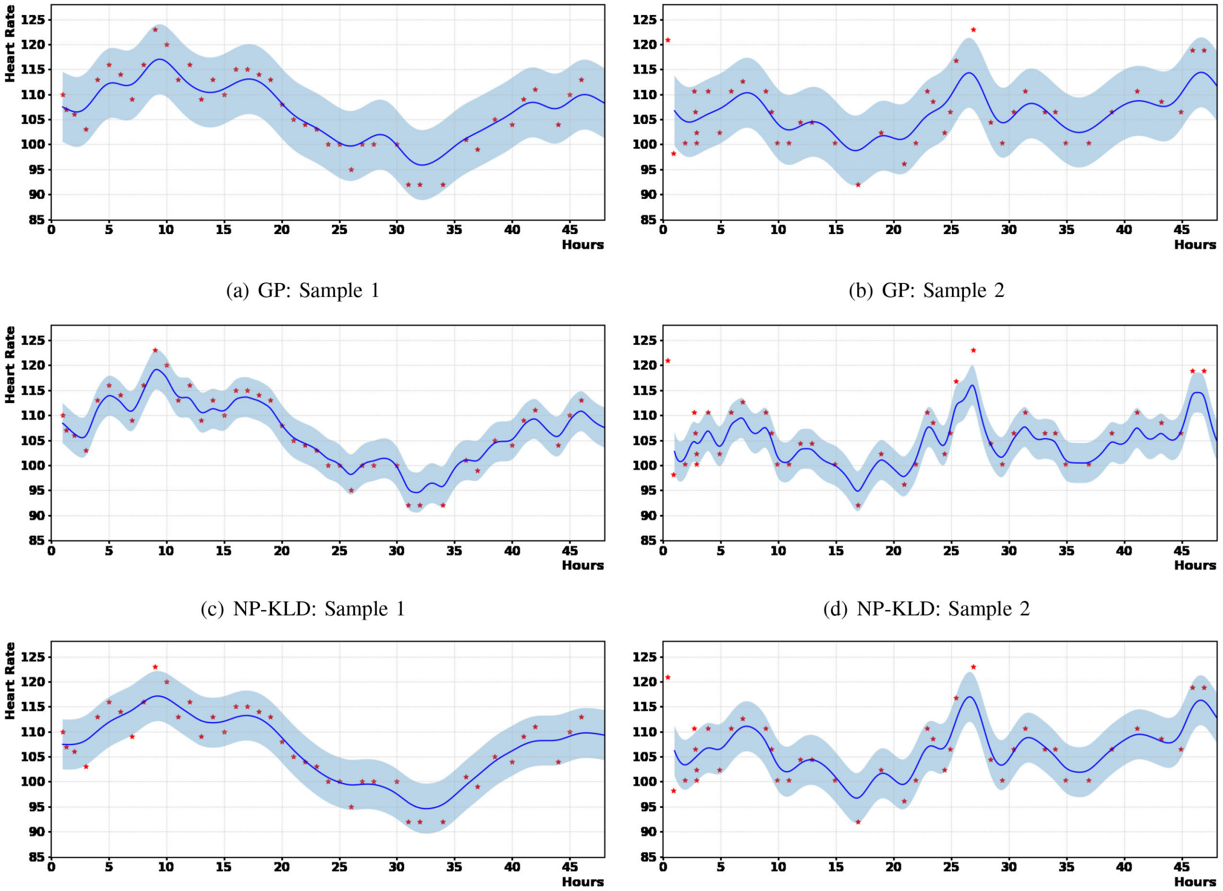
The CNN classifier consisted of three 1D convolutional layers each having 100 filters of size 3, stride 1 and zero padding. Note that the outputs of the second and third CNN layer have an effective resolution of a layer with filters of size 5 and 7, respectively. Hence, the outputs from these three CNN layers were concatenated together to extract the temporal features at different scales. The final features for classification were obtained by first using a global average pooling layer to aggregate feature maps followed by a dense layer of dimension 500.

3) Classifier 3:

We also experimented with a combination of CNN with LSTM where the multi-resolution output of the third CNN layer is fed to an LSTM layer to make predictions at the last time instance. In all models we use ReLU activation function (except the final fully-connected layer which employ tanh activation function), orthogonal initialization, and the Adam optimizer. All models were trained for 100 epochs using binary cross-entropy loss with a batch size of 8.

The models were also compared with a standard logistic regression classifier using the original data, as in [4] (labeled as LR), and the last value carry forward (CF) interpolation method. For LR based classifier, hand-engineered features as described in [36] are used. For each vital-sign, six different sample statistic features are computed on seven sub-sequences (time-series), these features being: maximum, minimum, mean, standard deviation, skew and number of measurements. The

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.



(a) GP: Sample 1

(b) GP: Sample 2

(c) NP-KLD: Sample 1

(d) NP-KLD: Sample 2

(e) NP-MMD: Sample 1

(f) NP-MMD: Sample 2

Fig. 2. Visualization of time-series modelling for vital signs for two patients using GP and NP models. (a), (c), (e) and (b), (d), (f) are GP/NP-KLD/NP-MMD modeling for two exemplar patients. The horizontal axis represents the time from admission and vertical axis represents the heart rate value. The original irregularly-sampled vital-sign measurements are shown with red markers (*) in the figure.

seven sub-sequences included the full time-series, the first 10% of time, first 25% of time, first 50% of time, last 50% of time, last 25% of time, and the last 10% of time. The CNN/LSTM classifiers using GP-, CF- and NP-interpolated data of vital signs are hereafter labeled as CNN/LSTM-GP, CNN/LSTM-CF and CNN/LSTM-NP, respectively.

V. Results

In this section, we compare the performance of the proposed NP-based data interpolation approach with the GP-based technique. Further, we evaluate the effectiveness of the GP- and NP-based interpolated data for different tasks considered in this work.

A. Data Interpolation

Here, we compare the performance of the proposed NP-based data interpolation model with KLD and MMD loss functions. In addition we also compare the NP-based model with the GP-based model. The experimental results for data interpolation experiments are shown using the train-test data for the in-hospital mortality task.

1) Np-Kld vs Np-Mmd: In contrast to the conventional KL divergence loss, the proposed approach uses the MMD loss for training NP. In order to demonstrate its effectiveness, we train two NP models using KL and MMD. Based on experimentation, $\alpha = 3$ was adopted for the KL loss and $\alpha = 100$ was adopted for the MMD loss, which are in line with the study presented in [21] for training such latent models.

As an illustration, Fig. 2 shows case studies of visualization of feature extraction from the original time-series data using GP and NP approaches for two different patients. This figure shows the mean and variance estimates obtained, where the horizontal axis represents the time from admission and the vertical axis represents the heart rate. We can observe that the NP-KLD model tends to over-fit more than the NP-MMD model. The NP-MMD model captured more variations while avoiding over-fitting. Such behaviour indicates that the MMD-based loss for NP prevents the model from over-estimating variance in clinical time-series data, so as to keep the latent space informative [21].

In order to estimate the modeling capabilities of the NP-KLD and NP-MMD methods, we estimated the root mean squared error (RMSE) on test data for different vital signs as summarized

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

TABLE I
MEAN AND STANDARD DEVIATION OF THE RMSE FOR DIFFERENT VITAL SIGNS USING CF, GP, AND NP WITH KLD AND MMD LOSS FUNCTION (FOR THE TESTING DATA)

Variable	**CF**	**GP**	**NP-MMD**	**NP-KLD**
HR	7.23±3.95	4.36±2.63	**3.16±2.37**	3.82±2.48
SBP	15.31±8.19	7.18±4.14	**6.03±3.19**	6.92±3.87
RR	5.14±2.58	3.27±1.96	**2.14±1.17**	2.90±1.87
DBP	15.43±9.39	7.28±4.53	**6.17±4.81**	6.89±4.03
SPO_2	3.84±3.37	2.38±2.35	**1.57±2.13**	1.93±2.18
MBP	14.32±6.72	7.93±3.92	**7.32±3.94**	7.91±3.91
TEMP	0.52±0.74	0.17±0.67	**0.14±0.44**	0.16±0.52
CRR	**0.21±0.37**	0.49±0.52	0.53±0.57	0.57±0.53
FIO	**0.15±0.32**	0.45±0.71	0.48±0.79	0.47±0.61
G-CSEO	**0.49±0.79**	0.61±0.83	0.73±0.81	0.74±0.85
G-CSMR	**0.73±0.76**	0.79±0.92	0.83±0.97	0.84±0.93
G-CST	**1.75±0.94**	1.85±1.25	1.89±1.37	1.92±1.47
G-CSVR	**0.63±0.81**	0.78±0.89	0.85±0.92	0.91±0.97
Glucose	**51.47±20.18**	52.16±30.26	53.79±35.52	54.83±37.29
pH	**1.59±1.93**	1.73±2.01	1.86±2.05	1.87±2.18
Weight	**11.49±6.73**	12.91±8.47	13.84±8.73	13.95±9.05

TABLE II
COMPARISON OF THE PROPOSED METHODS WITH VARIOUS APPROACHES FOR MORTALITY PREDICTION

Model	**AUROC**	**AUPRC**
SAPS [37]	0.720 (0.720, 0.720)	0.301 (0.301, 0.302)
APS-III [38]	0.750 (0.750, 0.750)	0.357 (0.356, 0.357)
OASIS [39]	0.760 (0.760, 0.761)	0.311 (0.311, 0.312)
SAPS-II [40]	0.777 (0.776, 0.777)	0.376 (0.376, 0.377)
LR [4]	0.848 (0.828, 0.868)	0.474 (0.419, 0.529)
LSTM-CF [4]	0.855 (0.835, 0.873)	0.485 (0.431, 0.537)
SAnD [41]	0.857	0.518
LSTM-GP	0.859 (0.841, 0.879)	0.496 (0.439, 0.546)
LSTM-NP	0.868 (0.846, 0.883)	0.509 (0.449, 0.554)
CNN-CF	0.849 (0.827, 0.854)	0.473 (0.428, 0.523)
CNN-GP	0.851 (0.824, 0.863)	0.486 (0.429, 0.533)
CNN-NP	0.854 (0.837, 0.871)	0.499 (0.439, 0.538)
CNN-LSTM-CF	0.864 (0.842, 0.884)	0.493 (0.427, 0.542)
CNN-LSTM-GP	0.869 (0.8497, 0.886)	0.509 (0.445, 0.559)
CNN-LSTM-NP	**0.875 (0.853, 0.895)**	**0.519 (0.454, 0.563)**

The bold values represents best AUROC and AUPRC for different methods.

in Table I. It can be observed that the mean RMSE is slightly lower for the NP-MMD as opposed to NP-KLD.

2) Np vs Gp: The visualization of feature extraction from the recorded vital-sign data using GP and NP approaches for two different patients is also shown in Fig. 2. It can be observed that the GP tends to learn a smoother curve with large variances which are ultimately capturing the general trend in vital-sign variations. On the contrary, the NP-based model with MMD loss seems to result in a curve which captures more variations than the GP-based model, while avoiding over-fitting.

The RMSE using GP-, NP and CF-based methods on the test data for different vital signs is summarised in Table I. We have not shown the results for height variable as this is highly sparse and is not likely to change for the patient. It can be also be observed that the mean RMSE for subset $\mathcal{V}$ is smaller for the NP-based models than it is for GP or CF. However, for the rest, CF method performs better, possibly because high sparsity in subset $\mathcal{U} - V$. Here, it is observed that the CF method performs better when the sparsity of the features is greater than 77%. When the sparsity is less than 77%, the NP model tends to follow closely the periodic/oscillatory trends in time-series. This also suggests that the NP will generally perform better than the GP as the number of training instances within a time-series window increases. NP as a machine learning model combines the inference speed of DNNs with predictive capability of GPs. Further, NPs are easy to implement and can scale to larger datasets and higher dimensions e.g., GP scales as $\mathcal{O}(n^3)$ compared to $\mathcal{O}(n)$ complexity in case of NP. As with any machine learning model (compared to direct imputation like carry-forward) the majority of computational load is governed by the training phase.

A trade-off between model performance versus amount of training data can be achieved by carefully optimizing the regularization loss in NP and we observed that NP with MMD loss achieves this. The major challenge lies in training an NP-based model with the best hyperparameter configuration, so as to obtain the best classification accuracy. However, this can be resolved using extensive architecture search. The modelling performance could be further improved using an end-to-end learning framework which is out of scope of this work. Similarly, other regularization loss functions proposed in context of various latent generative models could also be explored with NP, which we defer to future work.

B. Mortality Prediction

In this experiment, we evaluate the performance of deep learning classifiers with different interpolation techniques and compare the results with various existing methods. For a fair comparison, we follow the experimental setup of Harutyunyan *et al.* [4] where the data collected in the first 48-hour interval are used as input features. The metrics used to evaluate are the area under the receiver operator characteristic curve (AUROC) and area under the precision-recall curve (AUPRC). We compared the CNN/LSTM models with scores obtained with clinical methods, namely the Simplified Acute Physiology Score (SAPS) [37], Acute Physiological Score (APS) III [38], Oxford Acute Severity of Illness Score (OASIS) [39], SAPS II [40], and the single-task result in 'Simply Attend and Diagnose (SAnD)' [41].

The classification results are shown in Table II, where the NP models employ MMD loss. It can be observed that both LSTM-GP and LSTM-NP perform better than clinical methods, and better than the LR and CNN/LSTM-CF models that make use of the original raw data. This is as a result of efficient modeling of the vital-sign as opposed to simple CF interpolation. The mean AUROC for the combined CNN-LSTM-NP model is 0.875 which is the best among all the compared methods.

C. Decompensation Prediction

This experiment evaluates the performance of the proposed NP-based interpolation method on the decompensation prediction task. This task involves predicting a deterioration in the patients' condition in future. Thus, a binary label indicating a patient's death in next 24 hours is assigned to each hour of an ICU stay, after first four hours of admission to ICU. Following the existing setup, the number of instances in the train and test sets

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

TABLE III
COMPARISON OF THE PROPOSED METHODS WITH VARIOUS APPROACHES FOR DECOMPENSATION PREDICTION

Model	AUROC	AUPRC
LR [4]	0.870 (0.867, 0.873)	0.214 (0.205, 0.223)
LSTM-CF [4]	0.892 (0.889, 0.895)	0.324 (0.314, 0.333)
SAnD [41]	0.895	0.316
LSTM-GP	0.899 (0.892, 0.909)	0.332 (0.326, 0.340)
LSTM-NP	0.907 (0.893, 0.912)	0.342 (0.329, 0.351)
CNN-CF	0.883 (0.875, 0.891)	0.315 (0.302, 0.324)
CNN-GP	0.889 (0.879, 0.896)	0.325 (0.317, 0.331)
CNN-NP	0.897 (0.883, 0.908)	0.337 (0.323, 0.341)
CNN-LSTM-CF	0.899 (0.891, 0.903)	0.329 (0.319, 0.337)
CNN-LSTM-GP	0.905 (0.898, 0.912)	0.338 (0.329, 0.343)
CNN-LSTM-NP	**0.913 (0.903, 0.917)**	**0.345 (0.331, 0.353)**

The bold values represents best AUROC and AUPRC for different methods.

TABLE IV
COMPARISON OF THE PROPOSED METHODS WITH VARIOUS APPROACHES FOR PHENOTYPING

Model	Macro AUROC	Micro AUROC
LR [4]	0.739 (0.734, 0.743)	0.799 (0.796, 0.803).
LSTM-CF [4]	0.770 (0.766, 0.775)	0.821 (0.818, 0.825)
SAnD [41]	0.766	0.816
LSTM-GP	0.773 (0.769, 0.777)	0.821 (0.820, 0.825)
LSTM-NP	0.776 (0.771, 0.779)	0.825 (0.818, 0.827)
CNN-CF	0.769 (0.761, 0.771)	0.819 (0.816, 0.821)
CNN-GP	0.770 (0.768, 0.776)	0.820 (0.819, 0.825)
CNN-NP	0.773 (0.769, 0.779)	0.823 (0.820, 0.825)
CNN-LSTM-CF	0.771 (0.769, 0.777)	0.823 (0.819, 0.826)
CNN-LSTM-GP	0.775 (0.771, 0.779)	0.823 (0.820, 0.826)
CNN-LSTM-NP	**0.778 (0.772, 0.782)**	**0.826 (0.822, 0.829)**

The bold values represents best AUROC and AUPRC for different methods.

were 2,908,414 and 523,208, respectively (with decompensation rate of 2.06%) [4]. We use micro-average AUROC and AUPRC to measure the performance [4].

Results of this experiment are shown in Table III which are similar to those for the mortality prediction task, where we employ LSTM, CNN and CNN-LSTM classifiers. It is evident from the table that proposed NP-based interpolation technique performs better than GP- and CF-based techniques across different classifiers employed.

D. Phenotyping

In this experiment, various deep learning based classifiers are used to evaluate the performance for the phenotyping task. Here, the train and test data consists of 35,621 and 6,281 ICU stays with a full description of phenotypes described in [4]. Following the existing works in [4], [36], the results are reported in form of macro- and micro-averaged AUROC.

The data from the total ICU stay is used for this task, and the results are shown in Table IV. It can be observed that the GP-based interpolation yields better performance compared to the CF-based interpolation. Similar to the results for mortality and decompensation task, NP-based interpolation method outperforms other existing methods in this task as well.

VI. CONCLUSION

This is the first study to introduce the use of NP models to derive regularly-sampled data from highly-sparse and irregularly recorded physiological time-series data. We have demonstrated that the NP model can learn efficient representations for mortality prediction, decompensation prediction, and phenotyping classification tasks. The generative framework of the NP is regularized by maximizing a mutual information criterion between the distributions of context and all the training points. The use of this criterion maximizes the information between latent and input space, by ensuring that the latent conditional prior distribution does not have vanishing variances. The experimental results quoted earlier demonstrate the effectiveness of the proposed NP model on different tasks of the MIMIC III dataset.

The results in this paper demonstrate how neural network-based data interpolation methods may be preferred to conventional approaches. The percentage gain in AUROC over last value carry forward is rnage between 0.6%-1.6% for different predictive modeling tasks considered in this work. The gains are marginal and may not be clinically relevant. However, we would like to clarify that only seven out of total 17 variables are used for NP based interpolation (because of underlying data sparsity issues with rest of the variables). We expect to have further increment in performance if all the variables can be modeled using the proposed method.

In future, we aim to model jointly the temporal dynamics using a conditional generative approach and classification using a deep learning architecture. It would also be worth exploring the effect of various network architectural choices and optimization strategies on the overall cost function and the latent representations in the proposed NP framework.

ACKNOWLEDGMENT

The views expressed are those of the authors and not necessarily those of the NHS, the NIHR, or the Department of Health.

REFERENCES

[1] J. Henry, Y. Pylypchuk, M. T. Searcy, and V. Patel, “Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015,” 2015, *ONC Data Brief* 35. [Online]. Available: https://www.healthit.gov/data/data-briefs/adoption-electronic-health-record-systems-among-us-non-federal-acute-care-1.

[2] 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.

[3] J. Gao, C. Xiao, Y. Wang, W. Tang, L. M. Glass, and J. Sun, “Stagenet: Stage-aware neural networks for health risk prediction,” in *Proc. Web Conf.*, 2020, pp. 530–540.

[4] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan, “Multitask learning and benchmarking with clinical time series data,” *Sci. Data*, vol. 6, no. 96, pp. 1–18, 2019.

[5] A. Kumar and H. Anjomshoa, “A two-stage model to predict surgical patients’ lengths of stay from an electronic patient database,” *IEEE J. Biomed. Health Informat.*, vol. 23, no. 2, pp. 848–856, Mar. 2019.

[6] F. Ma, L. Yu, L. Ye, D. D. Yao, and W. Zhuang, “Length-of-stay prediction for pediatric patients with respiratory diseases using decision tree methods,” *IEEE J. Biomed. Health Informat.*, vol. 24, no. 9, pp. 2651–2662, Sep. 2020.

[7] B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel, “Unsupervised pattern discovery in electronic health care data using probabilistic clustering models,” in *Proc. ACM SIGHIT Int. Health Informat. Symp.*, 2012, pp. 389–398.

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.

[8] T. A. Lasko, J. C. Denny, and M. A. Levy, "Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," *PLoS One*, vol. 8, no. 6, 2013, Art. no. e66341.

[9] L. Clifton, D. A. Clifton, M. A. F. Pimentel, P. J. Watkinson, and L. Tarassenko, "Gaussian processes for personalized e-health monitoring with wearable sensors," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 193–197, Jan. 2013.

[10] M. Ghassemi, T. Naumann, T. Brennan, D. A. Clifton, and P. Szolovits, "A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 446–453.

[11] F. E. Shamout, T. Zhu, P. Sharma, P. J. Watkinson, and D. A. Clifton, "Deep interpretable early warning system for the detection of clinical deterioration," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 437–446, Feb. 2020.

[12] M. Scherpf, F. Gräßer, H. Malberg, and S. Zaunseder, "Predicting sepsis with a recurrent neural network using the MIMIC III database," *Comput. Biol. Med.*, vol. 113, pp. 103–395, 2019.

[13] R. Z. Wang, C. H. Sun, P. H. Schroeder, M. K. Ameko, C. C. Moore, and L. E. Barnes, "Predictive models of sepsis in adult ICU patients," in *Proc. IEEE Int. Conf. Healthcare Informat.*, 2018, pp. 390–391.

[14] X. Yang, Y. J. Kim, F. Khoshnevisan, Y. Zhang, and M. Chi, "Missing data imputation for MIMIC-III using matrix decomposition," in *Proc. IEEE Int. Conf. Healthcare Informat.*, 2019, pp. 1–3.

[15] A. Gelman and J. Hill, *Data Analysis Using Regression and Multi-Level/Hierarchical Models*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[16] M. Garnelo *et al.*, "Neural processes," in *Proc. ICML Workshop Theor. Foundations Appl. Deep Generative Models*, 2018. [Online]. Available: https://arxiv.org/abs/1807.01622.

[17] M. Garnelo *et al.*, "Conditional neural processes," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1704–1713.

[18] L. Clifton, D. A. Clifton, M. A. F. Pimentel, P. J. Watkinson, and L. Tarassenko, "Gaussian process regression in vital-sign early warning systems," in *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, 2012, pp. 6161–6164.

[19] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[20] G. W. Colopy, M. A. F. Pimentel, S. J. Roberts, and D. A. Clifton, "Bayesian Gaussian processes for identifying the deteriorating patient," in *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, 2016, pp. 5311–5314.

[21] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Information maximizing variational autoencoders," in *Proc. ICML Workshop Theor. Foundations Appl. Deep Generative Models*, 2018. [Online]. Available: https://arxiv.org/pdf/1706.02262.pdf

[22] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[23] M. Phuong, M. Welling, N. Kushman, R. Tomioka, and S. Nowozin, "The mutual autoencoder: Controlling information in latent code representations," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=HkbmWqxCZ

[24] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A Kernel method for the two-sample-problem," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2007, pp. 513–520.

[25] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1718–1727.

[26] V. Abrol, P. Sharma, and A. Patra, "Improving generative modelling in VAEs using multimodal prior," *IEEE Trans. Multimedia*, vol. 23, pp. 2153–2161, 2021, doi: 10.1109/TMM.2020.3008053.

[27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[28] R. Fortet and E. Mourier, "Convergence de la répartition empirique vers la répartition théorique," in *Proc. Annales Scientifiques de l'École Normale Supérieure*, vol. 70, no. 3, 1953, pp. 267–285.

[29] S. Vishwanathan, N. N. Schraudolph, and A. J. Smola, "Step size adaptation in reproducing Kernel Hilbert space," *J. Mach. Learn. Res.*, vol. 7, pp. 1107–1133, Jun. 2006.

[30] A. E. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, pp. 1–9, 2016.

[31] B. W. *et al.*, "National early warning score (NEWS): Standardising the assessment of acute-illness severity in the NHS," *Royal College Physicians*, 2012.

[32] F. Tang, C. Xiao, F. Wang, J. Zhou, and L.-w. H. Lehman, "Retaining privileged information for multi-task learning," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1369–1377.

[33] G. W. Colopy, "Bayesian Gaussian processes for identifying the deteriorating patient," Ph.D. dissertation, Dept. Eng. Sci., Univ. Oxford, 2018.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, [Online]. Available: http://www.deeplearningbook.org

[36] Z. C. Lipton, D. C. Kale, C. Elkan, and R. C. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016. [Online]. Available: https://arxiv.org/abs/1511.03677

[37] J. Gall *et al.*, "A simplified acute physiology score for ICU patients," *Crit. care Med.*, vol. 12, pp. 975–987, 1984.

[38] W. A. Knaus *et al.*, "The APACHE III prognostic system: Risk prediction of hospital mortality for critically ill hospitalized adults," *Chest*, vol. 100, no. 6, pp. 1619–1636, 1991.

[39] A. E. W. Johnson, A. A. Kramer, and G. D. Clifford, "A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy," *Crit. Care Med.*, vol. 41, pp. 1711–1718, 2013.

[40] J. Gall, S. Lemeshow, and F. Saulnier, "A new simplified acute physiology score (SAPS II) based on a European/North American multicenter study," *J. Amer. Med. Assoc.*, vol. 270, pp. 2957–2963, 1993.

[41] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4091–4098.

Authorized licensed use limited to: University of Oxford Libraries. Downloaded on June 01,2022 at 11:45:41 UTC from IEEE Xplore. Restrictions apply.