SIFT: A Deep Network for Irregularly-Sampled Multivariate Time Series

by

Aleksa Perucic

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

December 2020

APPROVED:

Professor Elke A. Rundensteiner, Thesis Advisor

Associate Professor Xiangnan Kong, Thesis Reader

Abstract

Due to the way hospitals record information, patient health records are composed of sparse and irregular multivariate time series data, with long breaks between visits and varying physiological variables recorded each time. Such data can provide valuable machine learning opportunities for improving patient care. Since standard Recurrent Neural Networks (RNNs) rely on fixed-length vector inputs, existing methods work by first converting irregular data into regular. However, such conversion requires a priori specifying a time interval to align values on, at the risk of negatively augmenting the original data and/or meaning of the data expressed by such irregularity. To address this open problem, we propose a novel end-to-end Reference-Network architecture (SIFT) for irregular multivariate time series to predict patient health outcomes. The architecture is composed of a Reference Network, an Interpolator Network, and a Discriminator Network. SIFT is able to extract the most informative data by adjusting the sampling interval to filter out noise and pass only the most useful signals to the classifier. SIFT adapts the sampling strategy (time interval data alignment) for values at which time points to interpolate, this way paying more or less attention to different time windows. The reference network is rewarded for sampling from the discriminative signal, and penalized for sampling from noisy data. We validate our approach on a range of recently proposed models, including GRU-D [1] and IPN [2]. Our experiments demonstrate that SIFT outperforms five comparable imputation and interpolation methods in various settings, in both AUC and Accuracy.

Acknowledgements

I would like to thank my advisor Professor Elke Rundensteiner, and my research colleague Tom Hartvigsen (Data Science PhD student at WPI). Their guidance and input over the past year has helped me learn an incredible deal about deep learning and data mining, and taught me good research practices. I would also like to thank James Kingsley of WPI Academic Research and Computing (ARC), for generously providing access to high performance computing resources used in this work.

Contents

Chapter 1: Introduction

1.1 Background and Motivation

The classification of time series data is an important part of various domains, such as healthcare [1, 3], meteorology [4], and financial marketing [5]. An important sub-set of time series classification deals with performing classification on irregularly sampled time series (ISTS). These time series are typically marked by irregularly-spaced gaps between observations, usually resulting from sensor errors, influence of experts and various other factors. For example, electronic health records (EHRs) record patient data such as physiological signals, medications, diagnostic codes, in-hospital mortality, length of stay, and are acquired by healthcare workers from doctors and nurses during routine hospital care [2]. This information provides valuable insights into the progression of a patient's health. Unfortunately, such data is marked by the lack of alignment in the observation times across various physiological variables, for instance the measurement of body temperature or blood pressure. EHRs can give important insight into how patients respond to different treatments and medications. By being able to interpret and make predictions on the multivariate irregular time series found in EHRs, doctors and healthcare workers would be able to forecast patient health outcomes without having to repeatedly record every physiological signal.

1.2 State-of-the-art

Recently, there have been several approaches to the problem of sparse and irregular time series data [1, 6, 7, 8, 9]. Arguably the simplest method for handling missing observations in time series classification is through imputation [10]. Imputation is defined as the process by which missing values are filled in [11], and the output of imputation is an evenly spaced regular time series. Several imputation strategies exist, but the most commonly used are zero (missing values replaced with zero), mean (missing values replaced with mean), and forward fill (missing values replaced with last real observation). While imputation can provide a straightforward and easy way to fill missing values, its simplicity can also be a limitation. Since the same imputed value is assigned to every missing value, this can lead to significantly distorted distribution and underestimated variance of imputed values. [11].

A more recent approach relies on constructing models with the ability to accept irregular time series as input [12]. Several recent frameworks have relied on the Gaussian Process (GP) [13, 7, 2]. Interpolation based on the GP allow for probability to be assigned to each estimated missing value. This is possible because of Gaussian distributions which are defined by covariance matrices and mean vectors [14]. These two elements in tandem allow for the derivation of the probability of an estimate. If we assume the mean is zero, the covariance function fully defines the behavior of the GP, and it becomes *non-probabilistic*. Methods relying on the probabilistic GP achieve much higher accuracy compared to simple methods like imputation, but they are computationally expensive and slow when used on large datasets.

The aforementioned methods of dealing with spare and irregular data have some shared limitations. The key limitation is that these methods require ad-hoc selection of a sampling rate at which to estimate values, which can significantly impact the classification performance by either adding high levels of estimation error, or removing fine-grained information. When the sampling interval is too long, the interpolation step is unable to capture fine-grained signals. This doesn't give the classifier enough useful information to work with. When the sampling interval is short, it increases the missing data rate. This forces the classifier to filter out noise, which can in turn reduce prediction accuracy.

1.3 Problem Definition

In this work we tackle the problem of performing accurate deep-learning classification from irregularly-sampled multivariate time series. Given an irregularly sampled time series, the goal of an interpolation model should be to sample the input data in such a way that it may be used by an RNN, while also preserving temporal information about the observations. Most commonly used sampling methods involve uniform sampling across the timeline, which depending on the length of the interval can either cause loss of temporal information or an increase in missing data. It should be feasible to identify points at which to sample the ISTS from, instead of relying on uniform sampling. By being able to identify and extract relevant sections of sequential data and only passing those into the model, we can retain temporal information without increasing noise caused by missing data. Additionally, being able to compress a highly dense ISTS into a few key points would intuitively point to improved classification accuracy, due to the classifier not having to filter out any unnecessary noise itself. The objective is defined as reducing number of reference timesteps (points that sampled from), while maintaining and possibly even improving accuracy.

1.4 Challenges

We summarize three major challenges of this problem as follows:

- *Noise/Signal Ratio*: In a variety of tasks, regions of time series with important signals are surrounded by noise which increases difficulty during network training. The classifier has to spend significant time learning to filter out this noise, which greatly increases training time and reduces overall accuracy.

- *Observation Timing*: In real-world datasets such as EHRs, variables are recorded at varying, non-overlapping intervals. The fact that variables are not updated with new observations at the same time can negatively impact models that don't consider changes across all variables when performing interpolation.

- *Uniform Sampling*: Commonly-used interpolation methods such as imputation rely on uniform binning with an adjustable sampling rate. The ad-hoc selection of a sampling rate can greatly influence accuracy, as coarse binning loses fine-grained signal, but dense binning introduces too much noise to the classifier.

1.5 SIFT

In this work, we present a novel deep learning architecture to the aforementioned challenges which we refer to as SIFT: **S**emi-parametric **I**nterpolation of **F**iltered **T**imesteps. The architecture is composed of two networks: a reference network and a predictor network. In principle, the predictor network may take many forms, as long as it takes a temporal sequence as input and performs classification on it. However, in line with related works [2, 15] we model this component as a Recurrent Neural Network (RNN) using Gated Recurrent Units (GRU) for our transition function. The reference network learns to focus interpolation on the most relevant parts of an irregularly-sampled time series with respect to the classification task, similar to an attention mechanism. By focusing on only the relevant parts of the time series, the reference network allows for the most discriminative information to be propagated to the predictor network, reducing its need to filter noise. Empirical studies on real-world tasks show that our approach outperforms baseline methods in a variety of settings.

The main contributions of our work can be summarized as follows:

- We propose a reference learner network that is able to identify regions of relevant signal in an input time series using deep learning. The reference learner network is designed to be integrated with an existing interpolation architecture.

- We incorporate the reference learner network into an architecture that relies on a Gaussian kernel to perform interpolation at the regions identified by the reference network, which are then used to classify the time series.

- We apply our method to one synthetic and two real-world time-sensitive time series classification tasks. The results of these experiments show that our method significantly outperforms comparable baselines in several key metrics. Most importantly, it gets much higher accuracy when using fewer estimated values.

Chapter 2: Related Work

2.1 Overview

Many models and architectures have been developed in recent years that handle sparse and irregular time series [1, 6, 7, 8, 9, 2]. The primary focus of many of these models is on EHR data. In order to model EHR data, deep learning architectures have been employed. These architectures have been developed Convolutional neural networks [16, 17], and plenty state-of-the-art models utilize Recurrent neural networks (RNNs). For example, several works utilize Long-Short Term Memory (LSTM) RNNs [18, 19, 12], and GRU networks are also commonly used [20, 8, 1, 21]. Models relying on RNNs aim to overcome the inherent limitation of how RNNs handle input data. Standard RNNs expect input as a sequence of tokens, which does not account for unequally spaced gaps between observations. This means that important temporal information is lost when working with ISTS. This problem is commonly addressed using imputation before the altered data is fed into a regular model.

2.2 Imputation

Imputation is the process of taking missing values in an input time series and filling them in according to some heuristic, usually for the purpose of retaining temporal information when using ISTS data on RNN-based models. The missing values are filled at a specified sampling rate. The smallest possible sampling rate at which you could fill missing values can be described as $\lim_{i \to 0} P_i$ where i is the interval at which imputed values P_i are sampled. However, fine-grained sampling rates are not only computationally inefficient, they also greatly increase the noise/signal ratio [15]. Using more realistic sampling rates can lead to undersampling, where two or more values occupy the same "bin" of imputed values, which can be solved by averaging the values inside the bin. Bins with no values are filled based on an imputation heuristic. The heuristic can be simple, such as filling in each missing value with the mean or with zero. These primitive heuristics can be quite powerful, as a recent work [1] used a GRU-D network combined with simple imputation heuristics that incorporated mask and time interval information into the GRU model. Missing variables would decay over time from the last observation to their empirical mean, in order to model the decreasing relevance of an imputed value.

More elaborate imputation strategies exist, such as kernel methods [22] and Gaussian processes [23, 24, 7]. For example, Ghassemi et al. (2015) successfully utilized multi-task Gaussian process models to perform regression and classification on sparse and irregular EHR data [23]. However, they noted the computational cost of Gaussian processes presents a significant limitation. A more recent work by Shukla et al. (2019) used a separate interpolation network based on Gaussian kernels to perform imputation [2]. Their interpolation network relies on a combination of fine-grain and coarse interpolations, and an intensity function, to model patterns in the timing of the observations. Clustering models, also known as mixture models, have been used to overcome problems associated with irregular data. These include Gaussian mixture models [9], and clustering with learned distance metrics has also been utilized [25].

2.3 Informative Missingness

Research has been done on treating missingness as an informative [26, 1, 8, 27, 19]. These approaches range from simply concatenating missing entries or timestamps to the RNN input, to creating specialized RNN models that incorporate masking and time intervals. Another approach that aims to tackle the same problem is utilizing attention mechanisms. For example, Ma et al (2017) use a combination of three attention mechanisms on the hidden state output of bidirectional RNNs [21]. Tan et al. (2019) presented a network that would utilize a time-aware GRU (T-GRU) that can directly analyze irregularly sampled data. It also utilized a dual-attention structure that would take into consideration both data-quality and medical-knowledge [15]. Pham et al. (2017) make a similar modification, albeit to an LSTM network, that allows it to directly parse irregular EHR data, while also embedding information about admissions and interventions to the vector space [19].

Chapter 3: Model Framework

We propose SIFT, which is the first ISTS classification architecture that can learn where to sample an input time series, separating regions of uninformative noise from regions of signal relevant to the classifier. This is accomplished by combining an attention mechanism with a separate interpolation network. The goal of SIFT is to provide a method that can achieve high accuracy while requiring significantly less reference timesteps to be sampled. A *reference timestep* refers to the limited amount of points an interpolation strategy chooses to sample values from. An interpolation strategy samples values from reference timesteps, usually for the purpose of using these sampled values to perform classification or regression.

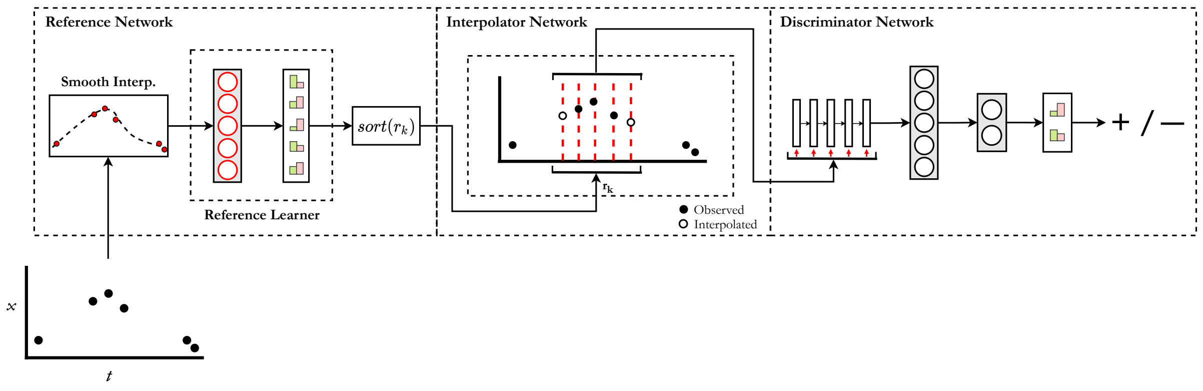

Figure 1 – SIFT: Model Architecture

As shown in Figure 1, the architecture of SIFT consists of three main components: a reference network, an interpolator network and a discriminator network. The reference network is comprised of a smooth interpolator, and a neural network we will from now refer to as the **R**eference **L**earner (**RL**). The smooth interpolator is used to make the initial interpolation, which the RL uses to select the most informative reference timesteps. The value at each reference timestep r needs to satisfy $0 \leq \mathrm{r} \leq 1$. Min-max scaling is used to map r between 0 and 1 . Through training, the reference network learns how to optimally pick out reference timesteps which contain discriminative information. The values at the reference timesteps are passed to an RNN-based classifier that learns a representation.

3.1 Reference Network

The goal of the reference network is to find discriminative regions of signal in ISTS data. The intuition behind this component is that certain "discriminative" regions of the input signal are more informative than others. Thus, given an input time series x_n and the number of reference timesteps to sample N, the network will return $N \times j$ points sampled from X_n, where j is the number of input features. These points are sampled from parts of the signal that the network believes are most informative. This is accomplished by training the **R**eference **L**earner (**RL**) component to perform accurate estimations of discriminative regions of signal on the input time series. The RL network consists of four sequential layers. A ReLU activation is applied to each layer but the last. Careful experimentation has shown that the RL learns best when a smoothed interpolation of the input signal is fed into it. Therefore, we introduce a smooth interpolator component, that uses a squared exponential Gaussian kernel with parameter α_t. The RL expects a single input vector x_d, which is the output of the smooth interpolator, and the RL performs reference timesteps selection independently across each input variable j.

The output of the reference network requires a certain property: For each reference timestep r_{jd}, the value at r needs to satisfy $0 \leq \text{r} \leq 1$. This is due to the fact that the both the timesteps and time series values of the input time series are normalized between 0 and 1. In order to guarantee the output of the network matches this property, min-max feature scaling is applied to the output of the final layer of the network h_t. Equation 1 shows the calculation of the final layer, while Equation 2 shows the min-max feature scaling applied to it. Due to how the network scales the input, it means that r_1 and r_k will always be 0 and 1, respectively. This results in the undesirable outcome where there are 2 fewer reference timesteps being drawn from the discriminative regions of the input time series. To prevent this from happening, we omit the maximum and minimum value, which are always 1 and 0 respectively, after sorting. (Equation 3). When training the SIFT architecture, the reference network is trained in tandem with the discriminator network. Thus, the output of the RL is a sorted vector $r \in [0, 1]$ that contains the set of *reference* timesteps.

$$h_t = xw_t + d \tag{1}$$

$$r_k = sort(\frac{h_t - h_{t,min}}{h_{t,max} - h_{t,min}}) \tag{2}$$

$$r_{k'} = (\hat{r_1}, r_2, r_3, ..., r_{k-2}, r_{k-1}, \hat{r_k}) \tag{3}$$

3.2 Interpolator Network

The interpolator network is inspired by the work of Shukla et al. (2019). The main benefit of this approach is that it provides a framework for easier interpolation across multiple variables. Their proposed Interpolation-Prediction Network (IPN) interpolates the irregular multivariate input time sequence against the reference points r_k provided by the reference network. More specifically, it expects an ISTS (comprised of pairs of observed timesteps t_d, and the observed values x_d) and the reference timesteps r_k to sample on. The IPN network initially creates two transformations of the input data: a Smooth interpolation σ_{kd}, and a Coarse interpolation γ_{kd}. The smooth and coarse interpolations are individually produced for each input feature j. The smooth interpolation uses a squared exponential Gaussian kernel $exp(-\alpha(r_{jk} - t_{jd})^2)$, with parameter α as α_d. The coarse interpolation uses the same squared exponential kernel with parameter α as $k\alpha_d$, where k is a parameter that adjusts the scaling of α. Thus, we can define the equations for smooth and coarse interpolation as follows:

$$w(r, t, \alpha) = exp(-\alpha(r - t)^2) \tag{4}$$

$$\sigma_{kd} = \frac{1}{\sum_{t \in t} w(r_{jk}, t_d, \alpha_d)} \sum_{j=1}^{L_{dn}} w(r_{jk}, t_{jd}, \alpha_d) x_{jd} \tag{5}$$

$$\gamma_{kd} = \frac{1}{\sum_{t \in t} w(r_{jk}, t_d, k\alpha_d)} \sum_{j=1}^{L_{dn}} w(r_{jk}, t_{jd}, k\alpha_d) x_{jd} \tag{6}$$

The interpolator network uses the smooth and coarse interpolations to provide three output vectors to the discriminator: the cross-channel interpolant captures smooth trends across variables, the transient component captures transients, and the intensity function captures temporal information about the observations.

3.3 Discriminator Network

The discriminator network takes the output from the interpolator and outputs a prediction of the target value for the given input vector. This component of the model is flexible and there are many state-of-the-art architectures for sequence classification. The output of the interpolation can be vectorized and fed into a standard RNN network. Several temporal and convolutional models such as GRU or LTSM can be used on time slices of the interpolator output. In this work, we conduct experiments leveraging a two-layer GRU network as the discriminator network due to its explicit modeling of the temporal dynamics present in irregularly-sampled time series and to remain comparable to prior work [2]. Each layer of the GRU computes functions defined by Equations 7-10.

$$q_t = \sigma(W_{ir}x_t + b_{ir} + W_{sr}s_{(t-1)} + b_{sr}) \tag{7}$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{sz}s_{(t-1)} + b_{sz}) \tag{8}$$

$$n_t = \tanh\left(W_{in}x_t + b_{in} + q_t \times W_{sn}s_{(t-1)} + b_{sn}\right) \tag{9}$$

$$s_t = (1 - z_t) * n_t + z_t * s_{(t-1)} \tag{10}$$

3.4 Optimizing SIFT

The process of training relies on iteratively updating all learnable parameters of SIFT, which is done by minimizing errors made by the Discriminator, and maximizing optimal reference timestep selection by the Reference Network. The Interpolator, Reference Network and Discriminator are optimized together with respect to Negative Log Likelihood loss defined in Equation 11, where y represents the prediction of the Discriminator as log probabilities, $\hat{y}$ the true label's probability, and K the number of classes of the prediction labels.

$$L(y) = -\frac{1}{K}\sum_{k=1}^{K} \hat{y}y \tag{11}$$

The final hidden state of the Discriminator, S_t, is projected into K-dimensional space using a three-layer fully-connected network. A softmax function is applied to the resulting vector, which allows the vector to be treated as prediction probabilities. Finally, we take the natural logarithm of the vector, since this operation has the effect of penalizing predictions that are confident and wrong. We can define the computation as follows:

$$P(Y = i|S_t, W_{ho}, b_{ho}) = softmax(W_{ho}S_t + b_{ho}) = \frac{e^{W_{ho}S_t+b_{ho}}}{\sum_j e^{W_{ho}S_t+b_{ho}}} \tag{12}$$

$$y = log(P(Y = i|S_t, W_{ho}, b_{ho})) \tag{13}$$

Where W_{ho} and b_{ho} represent the weights and biases of the three-layer network.

SIFT is an architecture composed of multiple components which are all optimized by a single loss function. In such cases, proper training of the model relies on the back-propagated gradients flowing from the loss function towards the beginning of the model, through each component. This process can work correctly as long as every operation is differentiable. Importantly, since the Reference Network uses a sort operation, we ensure this operation is differentiable by sorting the gradients alongside the logits.

Chapter 4: Experiments and Results

4.1 Baseline Models

For the baseline models, we compare our model to several imputation and interpolation methods that rely on a GRU classifier.

- GRU-M: Imputation strategy where missing values are replaced by the local mean. Multiple values occupying a bin are averaged.

- GRU-F: Imputation strategy where missing values are replaced by the the most recently observed real value. Multiple values occupying a bin are averaged.

- GRU-Z: Imputation strategy where missing values are replaced by the zero. Multiple values occupying a bin are averaged.

- GRU-D: Method proposed by Che et al. (2018) [1]. This method uses a modified GRU network which causes imputed values to decay over time to their empirical mean. The intuition behind this method is that as the time since the last real observation increases, the value of an imputed value decreases, matching a decrease in certainty that the imputed value is still representative of the actual data.

- IPN: Method proposed by Shukla et al (2019) [2]. This method uses a separate Interpolation Network that relies on Gaussian kernels to provide strong performance at performing interpolation across multiple variables. However, it still relies on uniform sampling of reference timesteps, which can limit accuracy in certain situations where signals are short, and reference timesteps are few.

To make baseline comparisons fair, only methods that utilize the concept of ”reference points” that are sampled on are included in the baselines. Therefore, methods where the entire input is passed, such as RNN-Δ_t, are excluded.

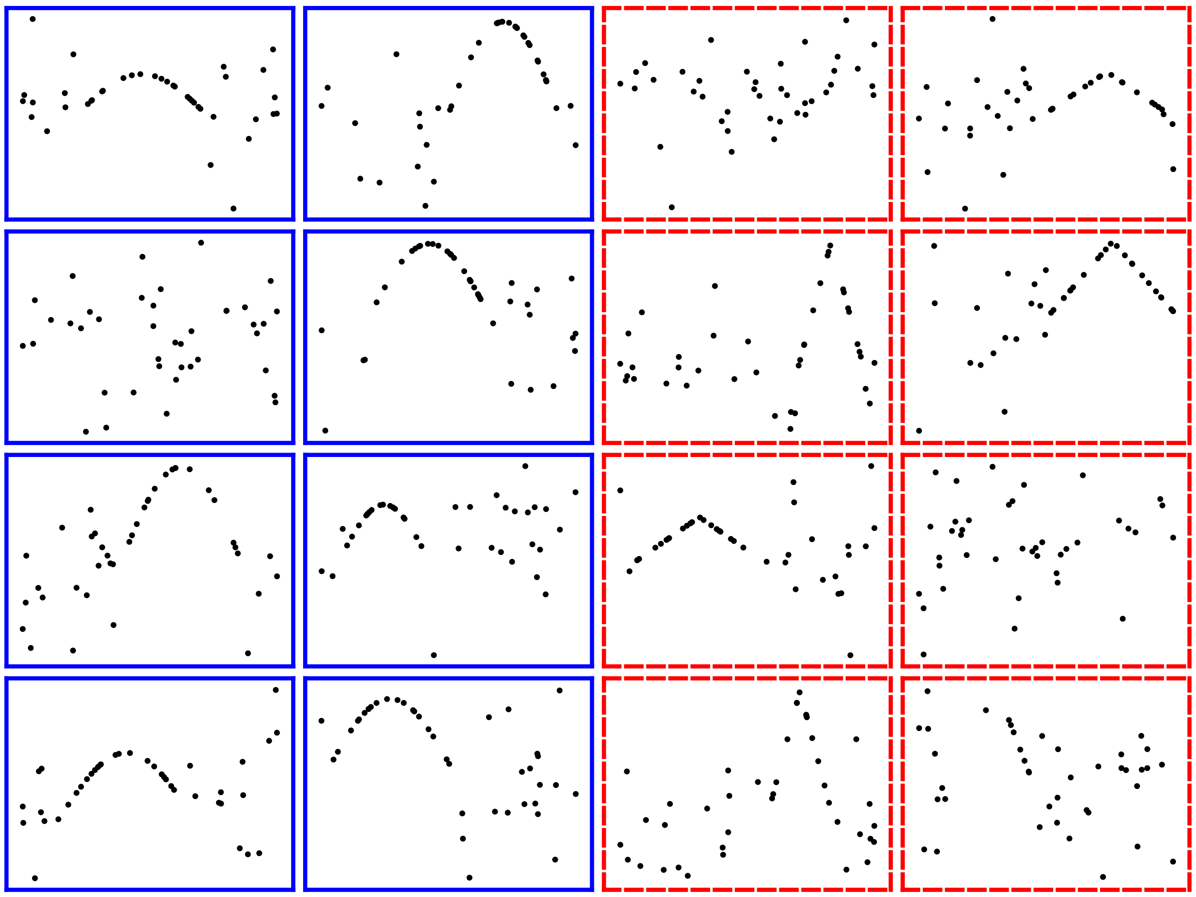

Figure 2 – Random samples from the Synthetic data with varying μ and λ. Solid-blue plots represent Class 0 (Domes), while dashed-red plots represent Class 1 (Spikes).

4.2 Datasets

4.2.1 Synthetic Data

The synthetic data-set was devised as a simple classification problem that could be made more difficult by adjusting its parameters. The data is defined as having two classes: domes and spikes. As seen in Figure 2, both classes have distinct shapes when plotted over the time axis.

While the problem is initially very easy, it can be made more difficult through two adjustments: noise (μ) and signal length (λ). Noise can be added either on the signal, or outside of the signal region. The addition of noise makes the problem more difficult by making regions that contain informative signal less discernible. The manipulation of signal length can also change the difficulty of the problem. By making the signal longer, interpolation methods that sample uniformly can pick more points which lie on the signal. This increases classification accuracy since more relevant points are provided to the discriminator network. By making the signal shorter, interpolation methods that sample uniformly will sample less observations from the signal region, leading to lower classification accuracy.

4.2.2 uWave

The uWave dataset is comprised of univariate recordings from the output of an accelerometer. The data contains 4000 samples of triaxial accelerometer data collected from eight users, with each recording having 94 observations. There are 8 possible classes, which signify 8 different gesture patterns identified by a Nokia research [28]. This data is irregular because an accelerometer only captures input during movement - meaning gaps in the data occur when the accelerometer is being held still.

4.2.3 PhysioNet

The PhysioNet dataset [29] has been widely used to evalute methods dealing with ISTS EHR data. The PhysioNet dataset is comprised of physiological recordings from the first 48 hours of 12,000 ICU stays. In order to remain faithful to related work of the baseline model(s), we remove features that don't positively contribute to classification accuracy [1]. Nine out of 41 features are chosen, and the selection is based on a recent work by Monterio et al. [30], which showed that removing certain features from the PhysioNet dataset can have a profound impact on accuracy, both in classification and regression. The data is normalized based on the minimum and maximum value across each time series. The data is front-padded with zeros in order to maintain the same sequence length across all features and patients. For the classification problem, a balanced sampling of set-a, set-b and set-c data is used for training. In order to fairly assess the performance of the model, validation and testing are performed on an imbalanced subset of the data.

4.2.4 Implementation Details

For all datasets, we split the data as 80% training and 20% testing. The training set is used to tune the model's parameters, while the testing set is used to evaluate certain hyperparameter configurations. In order to accurately represent the performance of the model and baselines, all results are averaged across 10 runs of the training process. In this work, we rely on the Adam method for optimization, with the learning rate adjusted according to the input dataset.

4.2.5 Experimental Setup

The experiments were performed on a linux platform, leveraging 13GB RAM, 2vCPU @ 2.2GHz, and an Nvidia K80 GPU with 2 cores and 12GB VRAM. The source code was written in Python 3.7.6, using libraries such as PyTorch 1.4.0, Numpy 1.17.3, and scikit-learn 0.22.2.

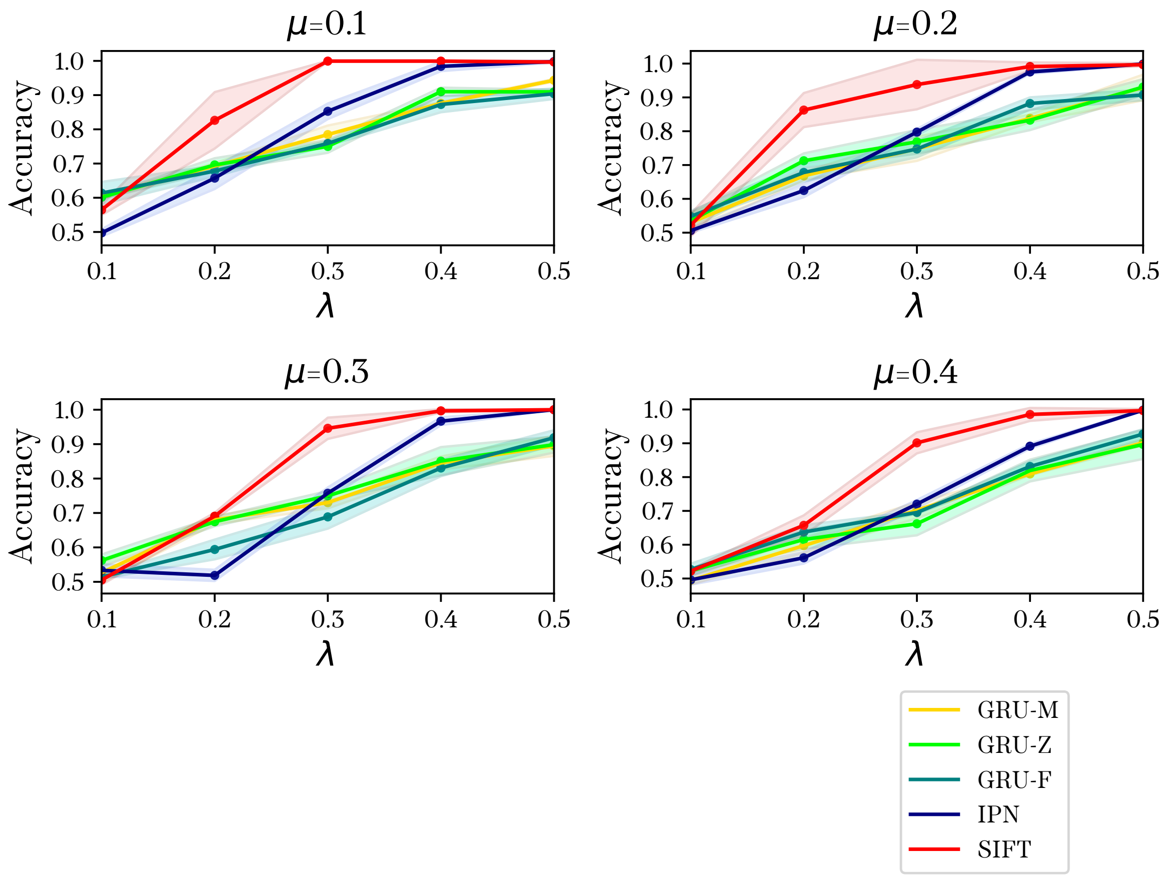

Figure 3 – Accuracy on Synthetic data while adjusting μ and λ. Confidence intervals represent standard deviation over 10 replications.

4.3 Experimental Results

4.3.1 Synthetic Data

The initial experiments are run on the synthetic dataset, which provides a controllable setting with clearly defined input data. Two important adjustable hyperparameters are found: the signal length λ, and the noise amount μ.

The hypothesis for the SIFT method is that it should be able to focus the interpolation entirely on the signal region, ignoring noisy and irrelevant data. We can test this using the synthetic data hyperparameters.

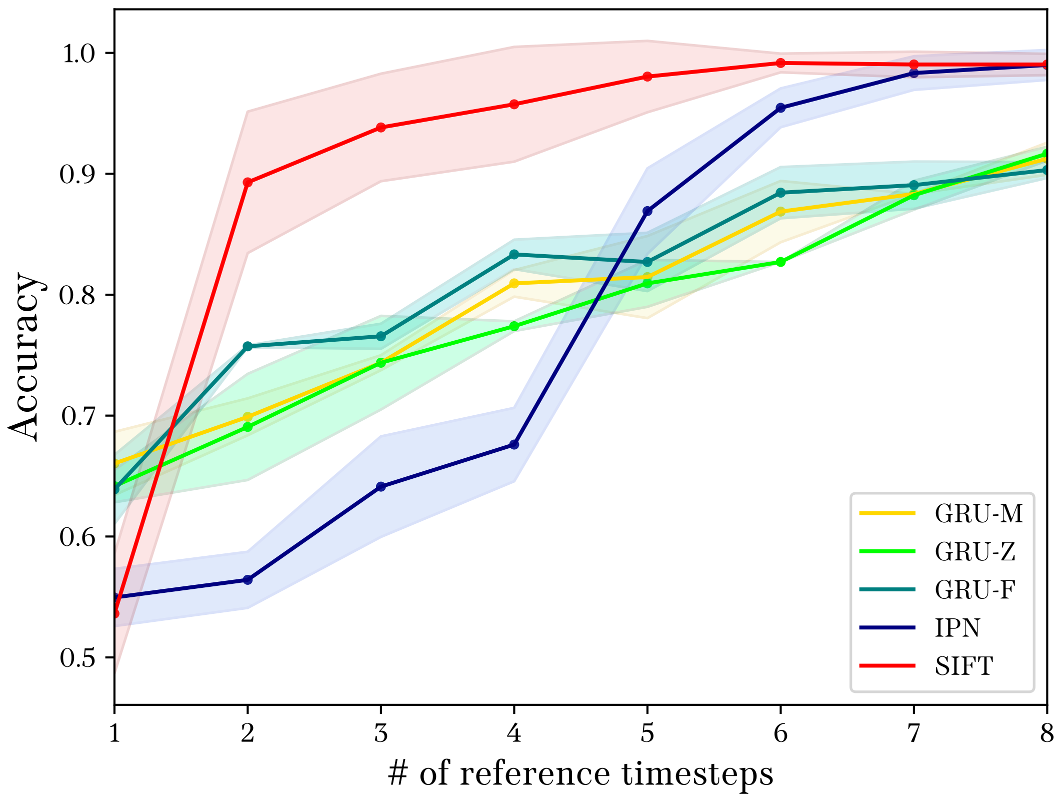

Figure 4 – Accuracy on Synthetic data with fixed λ and μ. Confidence intervals represent standard deviation over 10 replications.

Accuracy: SIFT should more accurately classify instances where less reference timesteps land on the signal. This means a shorter signal combined with fewer reference timesteps should provide the environment in which SIFT outperforms other methods. In order to test this, we fix the number of reference timesteps to 4, and adjust λ and μ. In Figure 3, we show how adjusting λ and μ plays a role in classification accuracy. We see that when λ is lower, and μ is higher, the classification task difficulty increases greatly. At $\lambda = 0.1$, all methods have approximately the same accuracy. This is because the signal is very small and difficult to dedect for any method, so methods get random accuracy. However, as the signal becomes more apparent as we increase λ, SIFT begins to outperform all the other baselines.

Figure 4 provides a comparison between SIFT's performance with a smaller λ, compared to baselines. As intuitively expected, SIFT seems to perform significantly better in situations where a uniform sampling of reference timesteps can't draw enough points from the signal. Additionally, increasing λ results in the expected behavior of equalizing the SIFT method with the baseline methods.

4.3.2 uWave

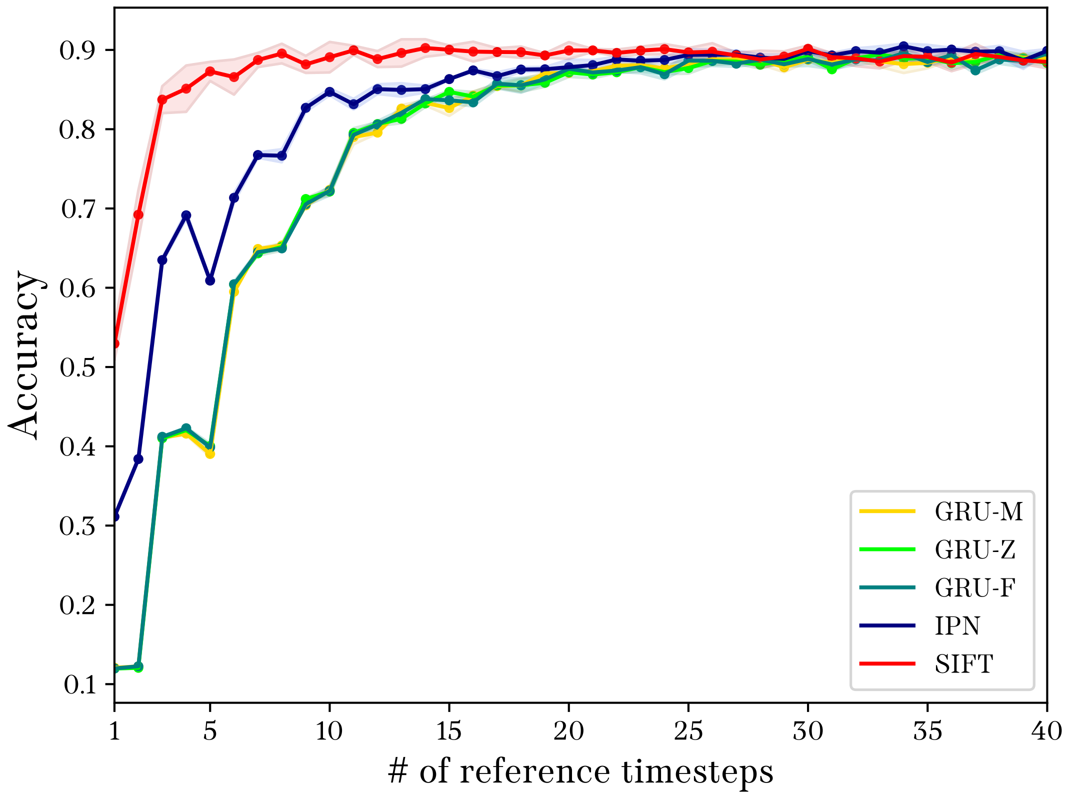

Figure 5 – Accuracy on uWave data while adjusting reference timestep count. Confidence intervals represent standard deviation over 10 replications.

The uWave dataset represents a univariate ISTS problem, which can be considered similar to the Synthetic data. Unlike the synthetic dataset, there are no parameters instrinsic to the dataset, so we can instead evaluate SIFT's performance by adjusting the number of reference timesteps. As mentioned several times throughout this work, we can expect that most methods will perform worse with less reference timesteps, and better with more. However, SIFT should be able to outperform other methods when using a smaller number of reference timesteps.

Accuracy: In Figure 5, we show that the results are in-line with the expected outcome. That is, SIFT performs tremendously well compared to other methods, when there are few reference timesteps. In fact, SIFT achieves over 50% accuracy with only 1 reference timestep on uWave, which is an 8-class classification problem. The baseline methods match SIFT's accuracy at 30 reference timesteps, but SIFT is already achieving the same accuracy at only about 15 - half the reference timesteps.

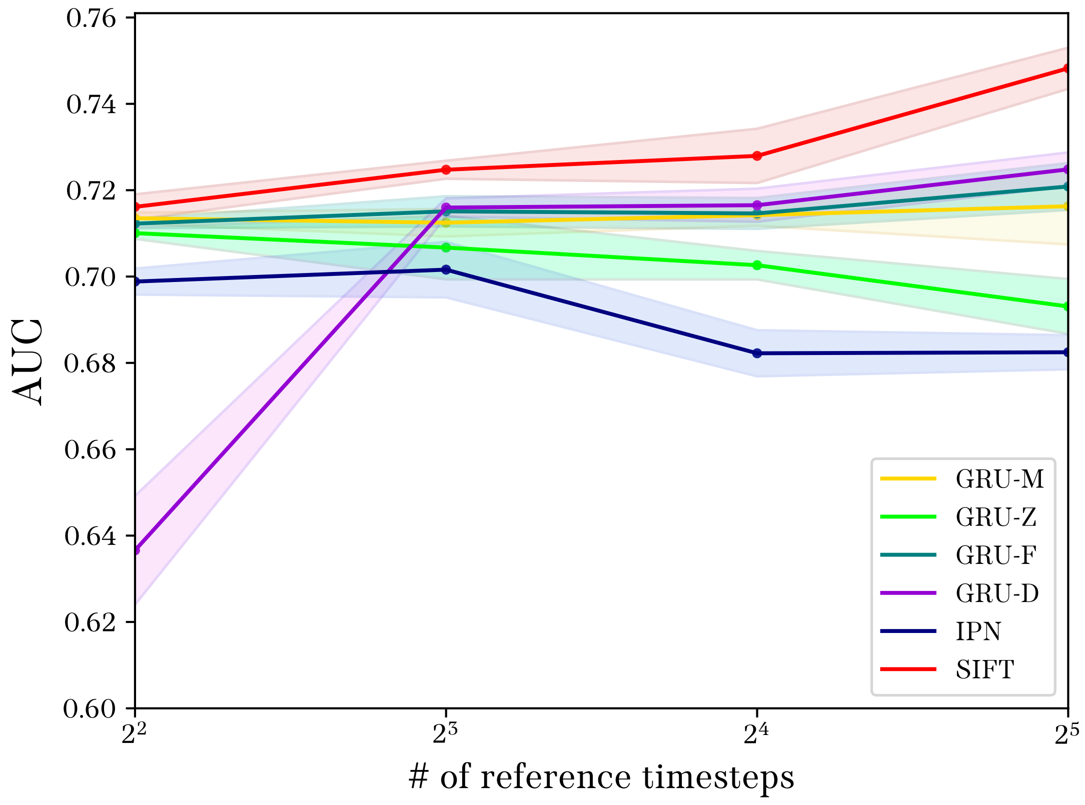

Figure 6 – Accuracy on PhysioNet data while adjusting reference timestep count. Confidence intervals represent standard deviation over 10 replications.

4.3.3 PhysioNet

PhysioNet is a multivariate dataset, in which the binary classification problem presents us with highly imbalanced data. In order to evaluate our method, we perform experiments in a similar style to the uWave ones. We adjust the number of reference timesteps, and evaluate how each baseline performs. Compared to the previous datasets, PhysioNet is a more complex and difficult problem by several orders of magnitude. Accordingly, we can expect the models to behave slightly differently than in the previous experiments.

AUC: In Figure 6, we show that SIFT still outperforms the baseline methods on the PhysioNet dataset. An interesting phenomenon is observed: the IPN and GRU-Z methods begin to perform worse as reference timesteps increase. This effect seems to validate our initial hypothesis: more data is not always preferable, if it is not being sampled from the correct signal region. This is further solidified by SIFT heavily outperforming the IPN baseline as we increase the reference timesteps, despite them both relying on a similar interpolation strategy.

Chapter 5: Discussion and Conclusions

5.1 Summary

In this work, we develop a novel architecture for the classification of irregularly sampled time series. Our Reference Network approach tackles limitations of existing imputation strategies on irregular time series data. We demonstrate that the addition of the Reference Learner component to an interpolation network can result in a more discriminative and informative interpolation. The Reference Learner accounts for the varying importance of different regions of an input signal, and adjusts the points of interpolation accordingly. This approach allows the classifier to learn more pertinent representations of the time series by filtering out data not useful for classification, with little performance overhead. Our experimental results on both synthetic and real-world datasets show statistically significant improvements in classification tasks over several relevant and comparable methods and architectures. Specifically, SIFT outperforms the baseline methods in situations where fewer points are sampled from an input time series, by being able to focus on only the relevant and discriminative regions of the input signal.

5.2 Future Work

This work is extensible in several directions. For example, the approach of the Reference Network is very similar to an attention mechanism. The implementation of the Reference Network relies on a straightforward and robust architecture consisting of an interpolator and a fully-connected four layer network (Reference Learner). However, novel approaches to attention mechanisms are published every year. It would be prudent to explore alternative implementations to the Reference Learner, by exploring the advances made in attention mechanisms. Additionally, the problem of this work can be viewed as an approach to dimensionality reduction. Our experiments show that using SIFT, input data can be compressed into lower dimensionality, while providing comparable or higher accuracy. This can be further explored as a way to perform data compression while maintaining discriminative regions of time series data.

References

[1] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Sci Rep*, vol. 8, p. 6085, 2018.

[2] S. Narayan Shukla and B. M. Marlin, "Interpolation-Prediction Networks For Irregularly Sampled Time Series," *CoRR*, vol. abs/1909.07782, 2019.

[3] Z. Liu and M. Hauskrecht, "Learning Linear Dynamical Systems from Multivariate Time Series: A Matrix Factorization Based Framework," *SIAM International Conference on Data Mining*, 2016.

[4] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, and H. Kong Observatory, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *CoRR*, vol. abs/1506.04214, 2015.

[5] S. Bauer, B. Schölkopf, and J. Peters, "The Arrow of Time in Multivariate Time Series," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 2043–2051, JMLR.org, 2016.

[6] S. Cheng, X. Li, and B. Marlin, "Classification of Sparse and Irregularly Sampled Time Series with Mixtures of Expected Gaussian Kernels and Random Features," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, p. 484–493, AUAI Press, 2015.

[7] J. Futoma, S. Hariharan, and K. Heller, "Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 1174–1182, 2017.

[8] Z. C. Lipton, D. C. Kale, R. Wetzel, and L. K. Whittier, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," *CoRR*, vol. abs/1606.04130, 2016.

[9] 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

Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, p. 389–398, Association for Computing Machinery, 2012.

[10] J. A. C. 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, 2009.

[11] P. Lodder, "To impute or not impute: That's the question," in *Advising on research methods: Selected topics*, Johannes van Kessel Publishing, 2013.

[12] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient Subtyping via Time-Aware LSTM Networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 10, p. 65–74, 2017.

[13] S. Cheng, X. Li, and B. Marlin, "A scalable end-to-end Gaussian process adapter for irregularly sampled time series classification," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, p. 1812–1820, 2016.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* The MIT Press, 2005.

[15] Q. Tan, M. Ye, B. Yang, S. Liu, A. J. Ma, T. C.-F. Yip, G. L.-H. Wong, and P. Yuen, "Data-gru: Dual-attention time-aware gated recurrent unit for irregular multivariate time series," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 930–937, Apr. 2020.

[16] F. Ma, J. Gao, Q. Suo, Q. You, J. Zhou, and A. Zhang, "Risk Prediction on Electronic Health Records with Prior Medical Knowledge," *KDD*, vol. 18, 2018.

[17] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh, "Deepr: A Convolutional Net for Medical Records," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, pp. 22–30, jan 2017.

[18] Z. C. Lipton, D. C. Kale, C. Elkan, and R. C. Wetzel, "Learning to diagnose with lstm recurrent neural networks.," *CoRR*, vol. abs/1511.03677, 2015.

[19] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Proceedings, Part II, of the 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume 9652*, PAKDD 2016, (Berlin, Heidelberg), p. 30–41, Springer-Verlag, 2016.

[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[21] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, (New York, NY, USA), p. 1903–1911, Association for Computing Machinery, 2017.

[22] K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurths, "Comparison of correlation analysis techniques for irregularly sampled time series," *Nonlin. Processes Geophys*, vol. 18, pp. 389–404, 2011.

[23] M. Ghassemi, M. A. F. 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 *AAAI* (B. Bonet and S. Koenig, eds.), pp. 446–453, AAAI Press, 2015.

[24] R. Dürichen, M. Pimentel, L. Clifton, A. Schweikard, and D. Clifton, "Multi-task gaussian processes for multivariate physiological time-series analysis.," *IEEE transactions on bio-medical engineering*, vol. 62, 08 2014.

[25] Z. z. Xing, J. Pei, and P. Yu, "Early classification on time series," *Knowledge and Information Systems - KAIS*, vol. 31, 04 2011.

[26] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 1555–1564, Association for Computing Machinery, 2016.

[27] E. Choi, M. T. Bahadori, and J. Sun, "Doctor AI: predicting clinical events via recurrent neural networks," *CoRR*, vol. abs/1511.05942, 2015.

[28] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "Uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mob. Comput.*, vol. 5, p. 657–675, Dec. 2009.

[29] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, "Predicting in-hospital mortality of ICU patients: The PhysioNet/Computing in cardiology challenge 2012," in *Computing in Cardiology*, vol. 39, pp. 245–248, NIH Public Access, 2012.

[30] F. Monteiro, F. Meloni, J. A. Baranauskas, and A. A. Macedo, "Prediction of mortality in intensive care units: a multivariate feature selection," *Journal of Biomedical Informatics*, vol. 107, p. 103456, 2020.