Selective imputation for multivariate time series datasets with missing values

Ane Blázquez-García, Kristoffer Wickstrøm, Shujian Yu, Karl Øyvind Mikalsen, Ahcene Boubekki, Angel Conde, Usue Mori, Robert Jenssen, and Jose A. Lozano

Abstract—Multivariate time series often contain missing values for reasons such as failures in the data collection mechanism. These missing values can complicate the analysis of time series data, and thus, imputation techniques are typically used to deal with this issue. However, the quality of the imputation directly affects the performance of subsequent tasks, especially when the missing rate is high. In this paper, we propose a selective imputation method that identifies a subset of time points with missing values to impute in a multivariate time series dataset. This selection, which will result in shorter and simpler time series, is based on both reducing the uncertainty of the imputations and representing the original time series as good as possible. The method is applied to different datasets to analyze the quality of the imputations and the performance obtained in subsequent tasks, such as supervised classification. The results show that it is not essential to impute all missing values as the optimal subset of time points can improve both the quality of the imputations and the accuracy of the classification.

Index Terms—Multivariate time series, missing data, imputation, irregular sampling

1 Introduction

A multivariate time series is a sequence of multivariate data points that have been recorded in an orderly fashion and are correlated in time. For instance, time series arise commonly in many application domains such as biology [1], astronomy [2, 3], geophysics [4], and health [4]. However, for reasons such as failures in the data collection mechanism, multivariate time series often contain missing values. Since the presence of missing values hinders an advanced analysis of time series data and complicates the subsequent application of machine learning algorithms for tasks such as classification or anomaly detection, their treatment is an important task to address.

Techniques in the literature usually tackle this issue using imputation methods. In general, we can categorize the imputation methods for time series into: 1) agnostic methods, which are defined as pre-processing methods and are independent of the subsequent machine learning task, 2) intrinsic methods, which are defined within the subsequent machine learning algorithm that will be applied.

Within the agnostic imputation methods, basic imputation techniques such as forward filling [5, 6], zero imputation [5], or mean imputation [6] have been widely used. More advanced techniques such as Generative Adversarial Networks (GAN) [7, 8] have also been proposed in this category. The main advantage of these techniques is that they can be used in combination with any machine learning task (e.g., forecasting, classification, or clustering) as they do not depend on the task itself.

In contrast, the intrinsic methods for multivariate time series are usually defined for classification tasks and use the information of the labels of the time series to impute the missing values [9, 10, 11]. As an example, Gaussian Processes have been used together with deep learning methods to obtain the imputed values [12, 13, 14]. Note that the imputations provided by these techniques are specific for the model and machine learning task (e.g., classifier) used.

In both the agnostic and intrinsic cases, the set of time points to impute needs to be determined beforehand. A naive solution is to impute all the missing values in all of the time points, assuming that the time series is regularly-sampled [10, 11, 15, 16, 17]. This solution is frequently adopted in the literature because many machine learning models require regularly-sampled time series without missing values (i.e., fully-observed time series) [18]. Indeed, authors often consider that the time series has an hourly sampling [5, 6, 12, 19]. However, these methods tend to make too many imputations; as an extreme example, they carry out imputations even in the time points where there is no measurement in any of the variables. Imputing so many missing values can produce high errors and affect the results of subsequent tasks, especially when the missing rate is high.

As such, more advanced techniques rely on imputing only the missing values in the time points where at least one of the variables has been observed [9, 20]. The resulting time series may have an irregular elapsed time between consecutive observations. Thus, for subsequent tasks such as classification, techniques in this group require choosing

- *A. Blázquez-García is with the Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA). Pº.J.Mª. Arizmendiarrieta, 2. 20500 Arrasate/Mondragón, Spain. E-mail: ablazquez@ikerlan.es.*
- *A. Conde is with Amazon Web Services. Torre foster,Paseo de la castellana 253a, 28046 Madrid, Spain.*
- *U. Mori and Jose A. Lozano are with the Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU. Manuel Lardizabal Ibilbidea 1, 20018, Donostia/San Sebastian, Spain.*
- *Jose A. Lozano is with Basque Center for Applied Mathematics (BCAM). Alameda de Mazarredo 14, 48009 Bilbao, Spain.*
- *K. Wickstrøm, S. Yu, K.Ø. Mikalsen, A. Boubekki, R. Jenssen are with UiT Machine Learning Group at the Department of Physics and Technology, UiT the Arctic University of Norway, Tromsø NO-9037, Norway.*

algorithms that are capable of dealing with irregularly-sampled time series. Although these techniques need to impute fewer values than in the previous case, it is questionable whether imputing all those data points is necessary to adequately represent the time series.

In this paper, we propose an agnostic method to selectively impute the missing values in a collection of multivariate time series, for the first time in the literature. In particular, the method selects the best subset of time points to impute based on the idea that selecting many time points can lead to a poor quality of the imputations, while selecting few time points can lead to a poor representation of the time series. In this way, the proposed method allows to shorten and simplify the time series, besides reducing both the error introduced by the imputations and the cost in different aspects (e.g., computational cost or the cost associated with the data collection).

The rest of this paper is organized as follows. Section 2 defines the context of the problem to be addressed and introduces the notation used throughout the paper. Section 3 presents the details of the proposed methodology. Section 4 provides the conducted experiments and the corresponding results. Finally, the conclusions drawn and suggestions for future work are discussed in Section 5.

2 Problem Setting and Notation

Let $D = \{Y^1, ..., Y^N\}$ be a time series dataset composed of N multivariate time series. Each time series Y^i is formed by L variables and contains missing values[1]. Additionally, let $\Omega = \{t_1, t_2, ..., t_T\}$ be the set of time points with at least one observation in D. An illustration of the problem setting is shown in Fig. 1, where the actual observations of each time series are represented by black crosses. In this example, it can be seen that D has observations in a total of eight time points (i.e., $|\Omega| = T = 8$).

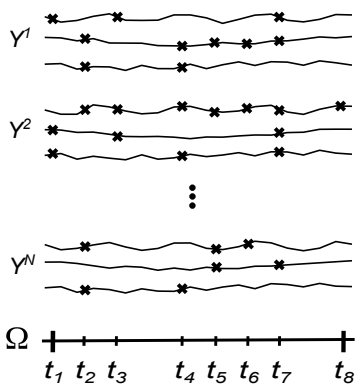


Figure 1: Illustration of the problem setting.

In this context, the main focus of this paper is to address the problem of imputing missing values of the multivariate time series in D. In particular, the objective of this paper is twofold: 1) selecting the optimal subset of time points, which we denote as P^*, where $P^* \subseteq \Omega$, and 2) filling the missing information on those time points.

For the sake of clarity, the notation used throughout this paper is summarized in Table 1.

Table 1: Summary of the notation used.

D	$\triangleq$	Time series dataset
N	$\triangleq$	Number of multivariate time series in D
Y^i	$\triangleq$	i^{th} multivariate time series in D
L	$\triangleq$	Number of variables of the time series in D
Ω	$\triangleq$	Candidate set of time points
T	$\triangleq$	Length of Ω
P	$\triangleq$	Subset of time points of Ω
P^c	$\triangleq$	Complementary set of P in Ω (i.e., $P^c = \Omega \setminus P$)
P^*	$\triangleq$	Optimal set of time points

3 Methodology

The overall diagram of the proposed methodology is shown in Fig. 2. The first step consists of imputing all the missing values in the candidate set Ω (see Section 3.1). Then, the criterion to evaluate the different sets of time points to impute is designed (see Section 3.2), and following this criterion, the optimal time points are identified (Section 3.3). Once the optimal set of time points of a time series dataset has been selected, the time series are represented by those time points (see the last step in Fig. 2), and the subsequent task would be performed using this reduced representation. The details of the methodology are explained below.

3.1 Imputation of the missing values

Since the time series in D contain missing values, the first step is to obtain imputations for all the time steps in Ω. For this, we use a probabilistic model, in particular, a Multi-task Gaussian Process (MGP) [21], not only to impute those values but also to provide the uncertainty of the imputations, as it will then help in assessing the quality of the imputations. This technique is useful to model multivariate time series because it considers the correlations between the variables of the time series. See Appendix A for more details on MGP.

In particular, an independent MGP will be fit to each of the multivariate time series in D. Given a multivariate time series Y^i, the corresponding model parameters will be learned using all its observed values. This can be seen as the first step of the pre-processing of the time series. Once the hyperparameters of the MGP model have been learned for each time series, we can obtain an estimated value together with its uncertainty for any missing time point in that time series.

3.2 Criteria for the time point selection

The next step consists of establishing a criterion to evaluate the quality of each subset of time points $P \subseteq \Omega$. For this purpose, it should be taken into account that selecting a subset of time points P implies, on the one hand, having to impute the missing values in P, and on the other hand, losing the actual observations that are not in this subset (i.e., observations in P^c).

1. Without loss of generality, in this paper, we assume that the dataset has multiple time series ($N > 1$) and variables ($L > 1$), but the method is also applicable to a single time series ($N = 1$) and/or univariate time series ($L = 1$).

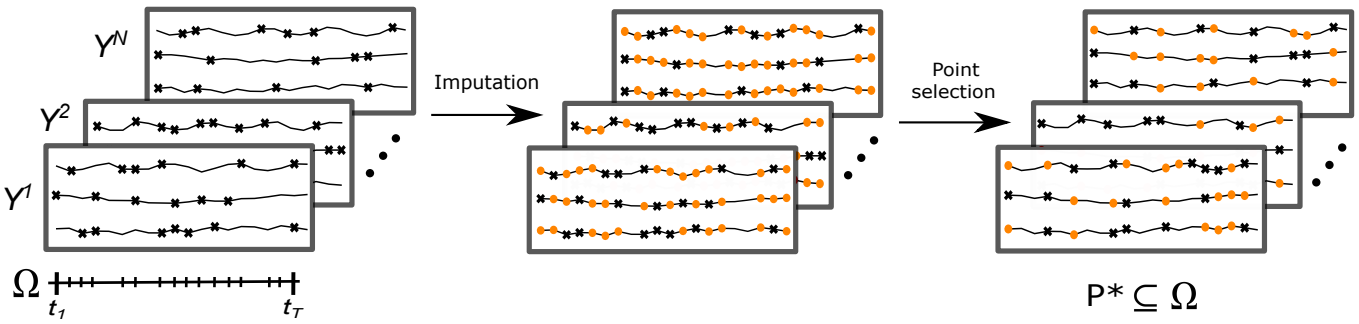


Figure 2: Diagram of the proposed methodology. Points in orange are the estimated values, and black crosses are the actual observations.

Therefore, the first criterion that we consider when evaluating a subset of time points P is the quality of the imputations of the missing values within P. To this end, we quantify the uncertainty of the imputations such that low uncertainty represents high imputation quality (see Section 3.2.1). On the other hand, we propose a second criterion to assess the quality of the actual observations within P that is based on measuring the information that is lost by excluding some of the time points. In particular, the more information that is lost, the worse the set of time points P is. To measure this, we introduce a new concept denominated predictive capability of a set of time points (see Section 3.2.2).

The details of the two criteria are described in the following sections.

3.2.1 Quantification of the uncertainty

As the imputations have been made with a probabilistic model, the uncertainty for a subset of time points P can be quantified using the variances of the imputations. In particular, we quantify the uncertainty in P of a time series i by computing the mean variance of the imputed values. That is,

$$V_P^i = \frac{1}{M_P^i} \sum_{j=1}^{M_P^i} \sigma_j^2 \tag{1}$$

where M_P^i is the number of missing values in time series i and set P, and σ_j^2 is the variance of the j^{th} imputed value. Note that this value is computed using the MGP.

To illustrate the intuition behind this criterion, an example is shown in Fig. 3. The aim is to quantify the uncertainty of the imputed values that are represented with orange dots. Specifically, the selection of points P consists of four time points and contains $M_P^i = 4$ missing values. Moreover, the uncertainty of their imputations is illustrated in blue by the confidence intervals of the predictions derived from the MGP. Based on this, in this example, the imputation of the missing value in the second variable has the poorest quality since it is the most uncertain.

Finally, in a collection of N time series, the best point set P^* in terms of this first criterion is the set of points that has the smallest uncertainty:

$$P^* = \arg\min_{P \subseteq \Omega} f_1(P) = \arg\min_{P \subseteq \Omega} \frac{1}{N} \sum_{i=1}^{N} V_P^i \tag{2}$$

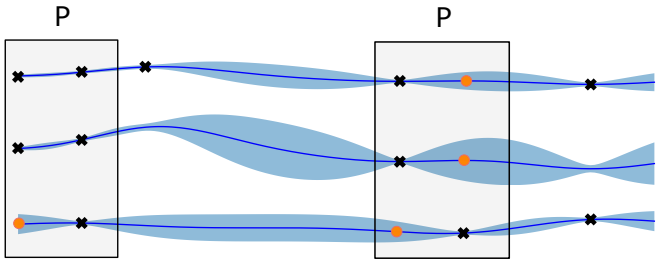


Figure 3: Illustration of the uncertainty of the imputed missing values within the set of time points P. The imputed values are shown with orange dots and the uncertainty with blue shading.

where $f_1(P) = \frac{1}{N} \sum_{i=1}^{N} V_P^i$ measures the overall mean uncertainty of the time series dataset for point selection P.

3.2.2 Quantification of the predictive capability

To measure the predictive capability of a set of time points P, a new MGP model is learned using only the actual observations in P. Then, we measure how well these points predict the observations that have not been included in P (see Fig. 4). The intuition is that if the points in P are able to predict the excluded observations accurately, then this exclusion is not causing a relevant loss of information.

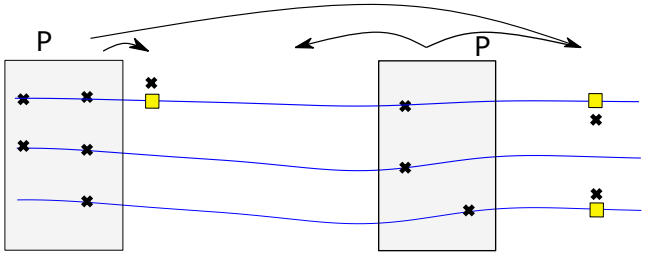


Figure 4: Illustration of the predictions of the excluded observations obtained using the observations in P. Actual observations are depicted by black crosses, and the predicted values of the excluded observations are shown by yellow squares.

To evaluate the predictive capability, we propose using

the Root Mean Squared Error (RMSE) in the following way:

$$PC_P^i = \begin{cases} \sqrt{\frac{1}{Q}\sum_{j=1}^{Q}(\hat{y}^i_{j,P^c} - y^i_{j,P^c})^2}, & \text{if } Q \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where Q is the total number of actual observations in P^c (note that $Q \geq 0$), y^i_{j,P^c} is the j^{th} actual observation outside P and time series i, and $\hat{y}^i_{j,P^c}$ is the respective predicted value that has been obtained using the observed values within set P. As an example, in Fig. 4, there are $|P^c| = 2$ time points that have not been selected, and there are $Q = 3$ actual observations within those time points P^c.

Finally, the best set of points P^* in a time series dataset consisting of N time series should obtain the maximum predictive capability globally, or, in other words, the minimum prediction error:

$$P^* = \arg\min_{P \subseteq \Omega} f_2(P) = \arg\min_{P \subseteq \Omega} \frac{1}{N}\sum_{i=1}^{N} PC_P^i \quad (4)$$

where $f_2(P) = \frac{1}{N}\sum_{i=1}^{N} PC_P^i$ measures the overall mean predictive capability in the time series dataset for point selection P.

3.3 Best sets of time points

The inclusion of many time points in P may involve having more missing values and a higher uncertainty of the imputations, but it also implies having a higher predictive capability since more actual observations are considered. On the contrary, the fewer points included in P, the fewer missing values there will be, having a smaller uncertainty, but also worsening the predictive capability because many observations are excluded. In general, uncertainty and predictive capability are conflicting objectives.

Thus, we formulate the problem of finding the best set of time points as a multi-objective optimization problem in terms of 1) uncertainty and 2) predictive capability:

$$\min_{P \subseteq \Omega} (f_1(P), f_2(P)) \quad (5)$$

The objective of this optimization is to find a Pareto set similar to the one that can be seen in Fig. 5. As illustrated in this figure, all the solutions in the Pareto set contain non-dominated solutions (i.e., subsets of time points), that is solutions that cannot be improved in any of the objectives without worsening the other objective. Note that this set dominates all solutions within the shaded region.

In particular, the two extreme solutions of the Pareto set in our problem are highlighted by green dots in Fig. 5. One of the extreme solutions corresponds to selecting all time points in Ω and is located at the bottom right in the figure (i.e., large f_1, and $f_2 = 0$). In this case, the prediction error is the minimum that can be obtained because no observations are excluded, while the uncertainty is very high since all missing values need to be imputed. Conversely, the other extreme solution, which is located on the top left of the figure (i.e., $f_1 = 0$, and large f_2), involves selecting a small set of time points in which no imputation has to be performed, and therefore, the uncertainty is 0 (i.e., the minimum that can be obtained). At the same time, this extreme set may contain very few time points and thus has the worst predictive capability because much information is lost and it is not able to reconstruct the excluded observations as well as other subsets in the Pareto.

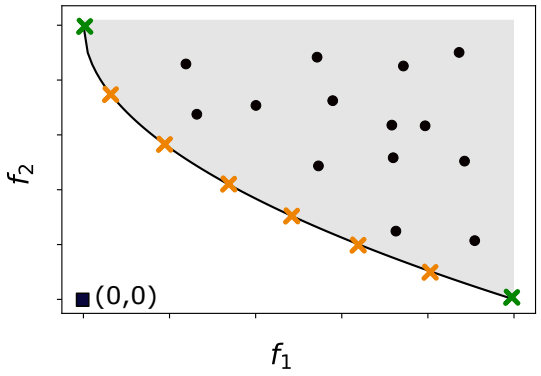


Figure 5: Example of a Pareto set illustrated by crosses. The extreme solutions in the Pareto are highlighted by green crosses.

Due to the large number of possible solutions (all possible subsets of Ω), we propose to use a meta-heuristic algorithm (e.g., NSGA-II [22]) to solve this multi-objective optimization problem. It should be noted that these algorithms do not necessarily reach the optimum but usually provide suitable solutions. Taking this into account, from this point on, we will refer to the sets in the Pareto as the optimal sets of time points but bear in mind that since we are using a heuristic, these solutions are an approximation of the Pareto.

4 Experiments

The experimentation is divided into two parts. The first part consists of analyzing the optimal sets of points P^* obtained by our method in synthetic datasets (see Section 4.1). In the second part, we apply our selective imputation method and analyze its performance when we apply a subsequent classification algorithm (see Section 4.2).

In both experiments, we assume that the missing behaviour in the dataset D is not random and that the time series share a common missing pattern. The reason for doing this is twofold. On the one hand, it will allow for a better interpretation and validation of the time point selection. On the other hand, in many time series datasets, missing data shares a common missing data pattern. For instance, in health data, patients admitted to the ICU that are progressing favorably and are not severely ill tend to receive less attention over time [23].

Parameter setting

The selected parameters for the MGP and the multi-objective optimization algorithm are common to both parts of the experimentation.

Concerning the MGP model, we use the `gpytorch` [24] library in Python and chose 100 iterations and a learning rate of 0.1. For the multi-objective optimization, we use the widely known NSGA-II algorithm [22], a multi-objective evolutionary algorithm that uses non-dominated sorting.

In particular, we use the `pymoo` [25] library in Python to implement this algorithm. This method has been selected based on its popularity, but since it is only an element of the framework, the evolutionary algorithm could be modified by the user. The specified parameters are the population size, which has been set to 20, and the number of generations, which has been set to 50. Additionally, we have initialized the algorithm such that the initial population contains the individual $P = \Omega$. The rest of the initial population is generated randomly.

4.1 Part I: Evaluation of the Pareto set in synthetic datasets

In this section, four different synthetic datasets are used to evaluate the performance of our approach in a controlled scenario. In all cases, the method is applied to a time series dataset of 100 bivariate time series.

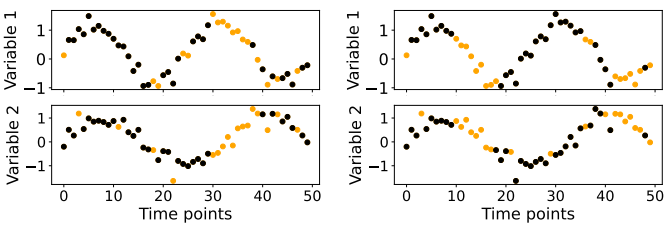
4.1.1 Generation of the synthetic datasets

The four synthetic datasets can be divided into two groups. The first group consists of two datasets generated using sinusoidal functions such that

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} sin((4\pi t)/T) \\ sin((3\pi t)/T) \end{bmatrix} + \begin{bmatrix} \xi_{1,t} \\ \xi_{2,t} \end{bmatrix}$$

where T is the length of the time series, $t \in \{0, ..., T\}$, and $[\xi_{1,t}, \xi_{2,t}]^T$ is the noise vector. For these experiments, we choose $T = 50$, and $corr(\xi_{1,t}, \xi_{2,t}) = 0.7$ to make $y_{1,t}$ and $y_{2,t}$ correlated. Moreover, for each $\xi_{i,t}$, given an interval $x^i = [x_1^i, x_2^i]$ with $x_1^i, x_2^i \sim N(0, 1)$ where the noise values will be, $\mathbb{E}(\xi_{i,t}) = \bar{x}^i$ and $Var(\xi_{i,t}) = (\sigma_{x^i}/3)^2$ where σ_{x^i} is the standard deviation of x^i. Then, missing values are injected such that most of the missing values are within a certain time interval A: the probability that each observation $y_{j,t}$ where $t \in A$ is missing is 0.9, whereas observations outside that interval have a probability of 0.2 of being missing.

The intervals chosen for conducting this experiment are $A_1 = [30, 40)$ and $A_2 = [10, 18) \cup [42, 48)$, each interval leading to a synthetic dataset in this group. An example of a time series in this group of synthetic datasets is shown in Fig. 6, for both of the intervals being analyzed. Based on the underlying idea of our proposal, we expect the method to avoid selecting points in A (A_1 in the first dataset, and A_2 in the second dataset), since this interval will have many missing values and, so, high uncertainty.



(a) $A_1 = [30, 40)$ (b) $A_2 = [10, 18) \cup [42, 48)$

Figure 6: Example of a time series in the first group of the synthetic datasets. The orange dots represent the missing observations.

The second group consists of two datasets generated based on a first-order Vector AutoRegressive (VAR) model such that

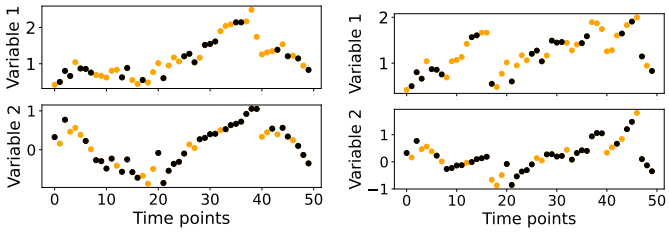
$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \xi_{1,t} \\ \xi_{2,t} \end{bmatrix}$$

where $t \in \{0, ..., T\}$. In particular, we choose $\alpha_0 = \alpha_1 = 0$, $\rho_1 = \rho_2 = 0.8$, and $T = 50$. Additionally, following [26], we choose the noise term such that $corr(\xi_{1,t}, \xi_{2,t}) = \rho(1 - \rho_1\rho_2)[(1 - \rho_1^2)(1 - \rho_2^2)]^{-1/2}$, where $corr(y_{1,t}, y_{2,t}) = \rho$ and $\rho = \rho_1 = \rho_2$. As in the previous group, given an interval $x^i = [x_1^i, x_2^i]$ with $x_1^i, x_2^i \sim N(0, 1)$ where the noise values will be, $\mathbb{E}(\xi_{i,t}) = \bar{x}^i$ and $Var(\xi_{i,t}) = (\sigma_{x^i}/3)^2$ where σ_{x^i} is the standard deviation of x^i. In this case, a particular time interval B is then replaced by a new, different process. This process consists of an increasing function such that for $t \in B$,

$$y_{i,t} = y_{i,t-1} + \epsilon_{i,t} \tag{6}$$

where $\epsilon_{i,t} \sim N(0, 0.2)$. Then, the missing values are injected uniformly throughout the time series with a probability of 0.4 of being missing.

As with the sinusoidal dataset, the intervals chosen for conducting the experiments are also $B_1 = [30, 40)$ and $B_2 = [10, 18) \cup [42, 48)$ (see Fig. 7). Note that each of these intervals also leads to a synthetic dataset in this group. In this case, our hypothesis is that the method will tend to select the time points in B, since this interval cannot be inferred by the points outside the interval.



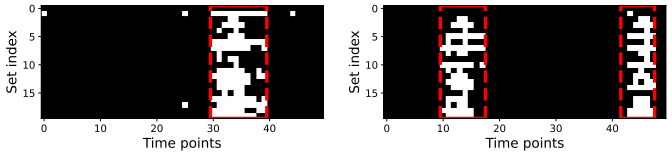
(a) $B_1 = [30, 40)$ (b) $B_2 = [10, 18) \cup [42, 48)$

Figure 7: Example of a time series in the second group of synthetic datasets. The orange dots represent the missing observations.

4.1.2 Results

The evaluation of the optimal subsets of points obtained by our method is performed in two parts: the first part analyzes the Pareto set in a qualitative manner, and the second part evaluates this Pareto by comparing it with randomly generated subsets of time points. In short, this section analyzes the results regarding the optimization part.

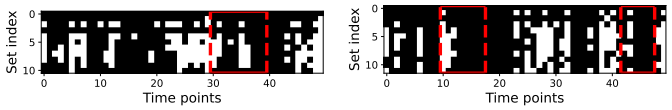
The selected subsets of time points for the two synthetic datasets in the first group are shown in Fig. 8. In particular, the black squares in this figure represent the time points that have been selected in each of the sets, and conversely, the white squares represent the time points that have not been selected. Also, the red lines highlight the intervals A_1 and A_2. As it can be seen in the figure, the most uncertain intervals (i.e., those with many missing values) are not selected: A_1 and A_2 contain most of the white squares.



(a) $A_1 = [30, 40)$ (b) $A_2 = [10, 18) \cup [42, 48)$

Figure 8: Optimal sets in the first group of synthetic datasets.

For the second group of synthetic datasets, the optimal subsets of time points are shown in Fig. 9. Unlike the first dataset, the method tries to include the intervals B_1 and B_2 as they provide new information that the rest of the points do not contain. In this case, the Pareto set contains fewer optimal sets than in the first synthetic dataset.



(a) $B_1 = [30, 40)$ (b) $B_2 = [10, 18) \cup [42, 48)$

Figure 9: Optimal sets in the second group of synthetic datasets.

On the other hand, to demonstrate that the point sets in the Pareto are good in terms of uncertainty and predictive capability, each optimal set is compared to 20 randomly generated sets of the same size. For instance, if an optimal set contains 15 time points, this set is compared to 20 randomly generated sets, each consisting of 15 time points. This analysis will help checking if the solutions are good enough because the optimization method used is heuristic. That is, we will examine if the optimization part has been performed adequately.

Specifically, for each set in the Pareto, this comparison analyzes, on the one hand, how many random sets dominate the set being analyzed (the number of random sets located in region 1 in Fig. 10), and, on the other hand, the set in question how many random sets dominates (the number of random sets located in region 2 in Fig. 10). It is desirable to have few points in region 1 and most of them in region 2. In particular, we perform this comparison in the cases in which the optimal set is not of length T, because otherwise all the random sets would be the same and the analysis would be meaningless.

For both groups of synthetic datasets, almost no random set dominates the corresponding optimal set (region 1 in Fig. 10). Particularly, in the first group, no optimal set is dominated by any random set, whereas, in the second group, there is only one random set that dominates the optimal set (on average, each set in the Pareto are dominated by 0.00% of random sets in scenario B_1, and 0.45% in scenario B_2). Conversely, when analyzing region 2, we find that the optimal sets dominate most of the random sets. In particular, the optimal sets dominate on average: in the first group, 95.26% and 96.58% of random sets in A_1 and A_2, respectively; in the second group, 55.50% and 71.36% of random sets in B_1 and B_2, respectively.

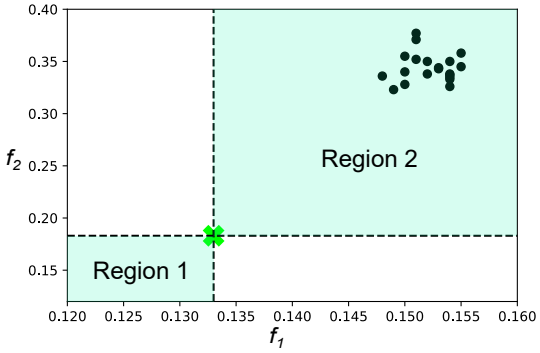


Figure 10: An example of the comparison between a set in the Pareto, which is depicted by a green cross, and 20 random sets of the same size, which are illustrated by black points.

4.2 Part II: Application in classification tasks

In this section, the usefulness of the proposed method in subsequent tasks such as multivariate time series classification is shown. We would like to emphasize that this section is not trying to demonstrate that our method is the best solution for the classification task but to illustrate that, with an appropriate selection of time points, it is possible not only to reduce the uncertainty and imputation error, but also to improve the results of the classification task.

As a preliminary proof of this hypothesis, in Fig. 11, we show the evolution of the mean accuracy of five popular classifiers when we perform a backward analysis in the *Libras* dataset [27] by removing the globally most uncertain time point of the time series dataset at each iteration. To obtain these accuracy values, we first pre-process each time series and impute all its missing values using MGP. The purple line in the figure indicates the accuracy obtained when using all the imputed time points, which corresponds to the 0^{th} iteration in Fig. 11. As can be seen, by removing the most uncertain time points from the time series, we can obtain an improvement in the results of the classifier up to eliminating approximately half of the time points.

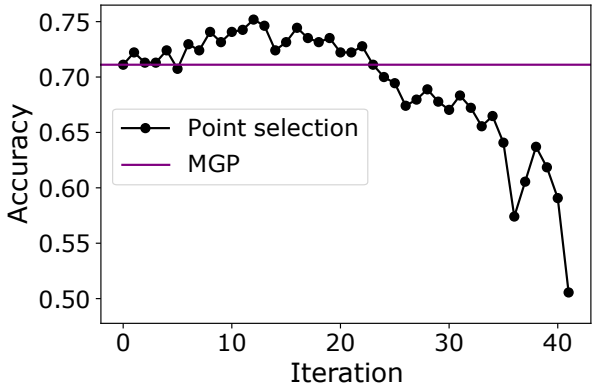


Figure 11: Backward analysis in the *Libras* dataset with 85% of injected missing data.

Now that we have seen that the selection of time points to impute can be beneficial for subsequent tasks such as multivariate time series classification, we will try to find the

optimal set of time points in different datasets and analyze the results of this task when using the simplified dataset.

4.2.1 Datasets

The experiments are performed in different datasets and classification tasks from the UEA repository [27]. Additionally, the dataset from the Physionet Challenge [28] that aims to predict in-hospital mortality is also used. In particular, for the Physionet dataset, we use a subset of samples, maintaining the mortality rate (14.29%), and variables, following [29], for efficiency and simplification.

The characteristics of the chosen datasets are summarized in Table 2, which shows the wide variety of the sets. In particular, for each case, 70% of the multivariate time series in the dataset is used to identify the best sets of time points and learn the classifier, and 30% for evaluation. Additionally, it should be noted that all the datasets described in the table originally have a regular sampling. We denote these equally-spaced time points as $X = \{1, 2, ..., T\}$, where T is the length of the time series.

While the Physionet dataset already contains missing values (it has an hourly sampling with missing values), the datasets from the UEA repository do not contain missing values. Thus, we inject the missing values in those datasets in such a way that the time series will contain more missing values at the end of the time series. In particular, if we denote mr_1 as the missing rate of the first half of the time series (i.e., $[1, ..., T/2]$), then $mr_1 \sim U(0.7, 0.8)$. In the same way, if we denote mr_2 as the missing rate of the second half of the time series (i.e., $[T/2+1, ..., T]$), then $mr_2 \sim U(0.9, 1)$.

Among the datasets used from the UEA repository, the *Japanese Vowels* dataset has time series of different lengths. In this case, a padding with missing values is made until the maximum length, which is 29, is reached. Then, the remaining missing values are injected to satisfy the missing rates described above.

The datasets used in this experimentation will be available in the GitHub repository[2] for further reproducibility.

4.2.2 Classifiers

Five traditional classifiers are used in the experimentation: Time Series Forest (TSF) [30], Mr-SEQL [31], 1-Nearest Neighbor using independent and dependant Dynamic Time Warping (DTW) distances [32], and RISE [33]. The score used throughout the experimentation is the mean accuracy of the five classifiers. For the classifiers that are designed to deal only with univariate time series (TSF, Mr-SEQL, and RISE), dimension concatenation is used [34]. The library used has been `sktime` [34] in Python, and the hyperparameters of the classifiers are set to the default values.

4.2.3 Baseline methods

Since techniques in the literature usually impute all the missing values, the baseline methods will be naive methods that will impute all the values of all the (equally-spaced) time points (i.e., X). In particular, the baseline methods will impute the missing values with the widely used Forward Filling (FF, baseline 1) and also with the Multi-task Gaussian Process (MGP, baseline 2). Once all the missing values have been imputed, all the time points will be used to learn the classifier.

2. https://github.com/ablazquezg

4.2.4 Results

In this section, we analyze both the quality of the imputations by computing the imputation error and also the accuracy of the classifiers using the sets of time points selected by our method.

To begin with, the imputation error is calculated in the test set using RMSE and normalized data between 0 and 1. In particular, the imputation error has only been computed in those datasets that originally have no missing values. As shown in Table 3, the imputation error is always smaller using the probabilistic MGP method than the FF method. Moreover, there are always sets in the Pareto that manage to reduce this error by using less time points. In particular, this reduction becomes very significant in some datasets, such as the *Libras* dataset.

The results regarding the classification accuracy are shown in Table 4. On the one hand, we analyze the results obtained using the baseline methods (i.e., when using all the time points), and we find that, in general, the MGP imputation provides better accuracy than the FF imputation. On the other hand, if we compare the accuracy results of the sets in the Pareto with those obtained with the baseline methods, we conclude that the proposed methodology is always able to find sets of time points that improve the accuracy (see columns $\geq FF(\%)$ and $\geq MGP(\%)$ in Table 4). Furthermore, the sets that fail to improve it manage to obtain similar results to the baselines but having reduced the number of time points significantly. For more details on the reduction of the time series, see Table 5. In general, the sets in the Pareto reduce the time series by an average of 27.12% of the length per dataset.

It should also be noted that in those cases in which the accuracy results of our method appear not to be successful, the imputation error is reduced. For example, the accuracy results obtained with the FF baseline method in the *Racket Sports* dataset are better than using our method. However, 96% of the sets in the Pareto obtain a lower imputation error than the FF baseline.

5 Conclusions and Future Work

In conclusion, this paper introduces a time point selection method to selectively impute the missing values in a multivariate time series dataset. This selection is based on the uncertainty of the imputed values and the predictive capability of the selected observations. In this way, the overall uncertainty of the dataset is reduced, and it allows to use simplified time series in subsequent tasks.

It should be noted that our method is not restricted to obtaining time series with equally-spaced time points, but this restriction could be added if desired. This makes our method more flexible than other techniques in the literature as it has the possibility to obtain time series with equally-spaced time points, where traditional machine learning techniques can be then applied.

The imputation method used to fill in the missing values has been MGP, but other probabilistic models that provide the uncertainty of the imputed values could also be used.

Dataset	Length	Dimensions	# of instances	Classes
Japanese Vowels	29	12	640	9
Racket Sports	30	6	303	4
Libras	45	2	360	15
Physionet	48	6	700	6
Finger Movements	50	28	416	2
Basic Motions	100	6	80	4
Epilepsy	206	3	275	4

Table 2: Description of the datasets used in the classification task.

Dataset	Baseline methods		Point selection				
	FF	MGP	min	mean	max	$\leq$ FF (%)	$\leq$MGP (%)
Racket Sports	0.2869 (0.0065)	0.2362 (0.0014)	**0.2362** (0.0014)	0.2608 (0.0016)	0.2836 (0.0023)	96.00 (4.18)	5.00 (0.00)
Libras	0.2974 (0.0038)	0.2094 (0.0050)	**0.1054** (0.0059)	**0.1654** (0.0038)	**0.2094** (0.0050)	100.00 (0.00)	99.00 (2.24)
Finger Movements	0.2845 (0.0013)	0.2664 (0.0019)	**0.2318** (0.0042)	**0.2576** (0.0036)	0.2688 (0.0033)	100.00 (0.00)	84.11 (10.00)
Basic Motions	0.2772 (0.0075)	0.1967 (0.0025)	**0.1963** (0.0029)	0.1978 (0.0028)	0.2003 (0.0031)	100.00 (0.00)	26.00 (16.36)
Epilepsy	0.3067 (0.0017)	0.2246 (0.0011)	**0.2204** (0.0018)	**0.2230** (0.0013)	**0.2246** (0.0011)	100.00 (0.00)	99.00 (2.24)

Table 3: Results of the imputation errors. The columns related to the baseline methods indicate the average imputation error with the standard deviation between parentheses of 5 different train/test partitions. The next three columns show some statistics of the imputation errors of the sets in the Pareto. The last two columns describe the percentage of the sets that achieve a lower imputation error than the baselines.

Dataset	Baseline methods		Point selection				
	FF	MGP	min	mean	max	$\geq$ FF (%)	$\geq$MGP (%)
Japanese Vowels	**0.7613** (0.0128)	0.7502 (0.0133)	0.7098 (0.0154)	0.7343 (0.0121)	0.7542 (0.0120)	14.89 (21.02)	16.78 (18.91)
Racket Sports	**0.4954** (0.0198)	0.4440 (0.0221)	0.4193 (0.0307)	0.4514 (0.0232)	0.4792 (0.0235)	3.00 (4.47)	69.00 (18.51)
Libras	0.5111 (0.0276)	0.6963 (0.0216)	0.6822 (0.0249)	**0.7022** (0.0143)	**0.7315** (0.0132)	100.00 (0.00)	65.00 (25.74)
Physionet	0.8215 (0.0088)	0.8276 (0.0042)	0.8168 (0.0088)	0.8240 (0.0060)	**0.8309** (0.0051)	42.22 (46.80)	62.22 (51.88)
Finger Movements	0.5251 (0.0154)	0.5245 (0.0146)	0.4925 (0.0136)	0.5196 (0.0139)	**0.5450** (0.0096)	50.11 (34.00)	50.33 (29.68)
Basic Motions	0.7483 (0.0272)	0.7933 (0.0266)	0.7700 (0.0240)	**0.7955** (0.0243)	**0.8233** (0.0239)	83.00 (38.01)	72.00 (8.37)
Epilepsy	0.8106 (0.0037)	0.8607 (0.0098)	0.8429 (0.0055)	0.8583 (0.0082)	**0.8713** (0.0145)	100.00 (0.00)	50.00 (28.94)

Table 4: Accuracies in the classification task. The columns in the table follow the same rationale as Table 3.

Dataset	Length max accuracy	Mean length	Mean reduction (%)
Japanese Vowels	19.60 (4.98)	16.11 (1.30)	44.46
Racket Sports	17.20 (5.26)	21.62 (1.17)	27.93
Libras	26.20 (4.27)	32.70 (1.50)	27.33
Physionet	33.40 (5.94)	37.06 (2.26)	22.79
Finger Movements	36.80 (13.25)	32.72 (0.90)	34.56
Basic Motions	71.8 (16.39)	81.79 (3.38)	18.21
Epilepsy	178.60 (21.76)	176.02 (4.85)	14.55

Table 5: Length reduction using the sets in the Pareto. The columns describe 1) the dataset used, 2) the lengths of the sets that provide the maximum accuracy, 3) the average length of the sets in the Pareto, and 4) the percentage reduction of this average. The values shown are the mean values of the 5 partitions and the standard deviation between parenthesis.

Since we use a probabilistic imputation method, an interesting future line of research could be to provide more sophisticated measures of uncertainty (e.g., using information theory). Moreover, in some contexts (e.g., when learning normality), it may be interesting to learn a single global imputation model on the whole dataset.

Reducing the set of time points does not only simplify the time series but can also help to improve the results of subsequent tasks such as whole time series classification. There are always sets in the Pareto that improve the accuracy, and those that do not improve it remain with a similar performance, but using a shorter representation of the time series. However, the use of more sophisticated classifiers could help to improve the results in terms of accuracy. In this line, future research could focus solely on improving the results of the classification task.

As mentioned throughout the paper, a significant advantage of the proposed method is that, in addition to the multivariate time series classification task, this method can also be used in combination with other subsequent machine learning tasks such as anomaly detection, forecasting, or clustering. Thus, an interesting line for future work would be to test the applicability of the method in additional subsequent tasks since the literature has mainly focused on the classification task.

ACKNOWLEDGMENTS

This research is supported by the Basque Government through the BERC 2022-2025 program and by Spanish Ministry of Economy and Competitiveness MINECO through BCAM Severo Ochoa excellence accreditation SEV-2017-0718, as well as through project TIN2017-82626-R funded by (AEI/FEDER, UE) and acronym GECECPAST. In addition, by the Research Group IT1504-22 programs (Basque Government), PID2019-104966GB-I00 (Spanish Ministry of Economy, Industry and Competitiveness) and Elkartek projects 3KIA (KK2020/00049), SIGZE (KK-2021/00095) and ALUSMART (KK-2021/00065).

APPENDIX A
MULTI-TASK GAUSSIAN PROCESS

Multi-task learning is a machine learning framework that aims to improve performance through the learning of multiple tasks at the same time, and sharing the information of each task [35]. Thus, Multi-task Gaussian Process (MGP) is an extension to Gaussian Processes (GPs) for handling multiple outputs at each time [21].

The objective of MGP is to model a set of processes $\{f_l(\boldsymbol{x})\}_{l=1}^{L}$, each one associated with a task, rather than a single process $f(\boldsymbol{x})$. When dealing with multivariate time series, the tasks refer to the dimensions of the time series (i.e., having L tasks means that the time series is L-dimensional). For convenience, we ignore the i^{th} superscript of the time series Y^i and use Y to refer to a time series in D. Additionally, we use $\tilde{T}$ to define the length of time series Y.

Given a set $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_{\tilde{T}}\}$ of $\tilde{T}$ indexes, the set of responses for L tasks is defined as the flatten vector $\boldsymbol{y} = [y_{11}, ..., y_{\tilde{T}1}, y_{12}, ..., y_{\tilde{T}2}, ..., y_{1L}, ..., y_{\tilde{T}L}]^T$, where y_{il} is the response for the l^{th} task on the i^{th} input $\boldsymbol{x}_i$. The observations are assumed to be noisy, and thus, each y_{il} is defined as

$$y_{il} = f_l(\boldsymbol{x}_i) + \epsilon_{il} \tag{7}$$

where $\epsilon_{il} \sim \mathcal{N}(0, \sigma_l^2)$. This can also be denoted as a $\tilde{T} \times L$ matrix:

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{\tilde{T}1} \\ \vdots & \ddots & \vdots \\ y_{1L} & \cdots & y_{\tilde{T}L} \end{bmatrix} \tag{8}$$

such that $\boldsymbol{y} = vec(Y)$. Each l^{th} row indicates the l^{th} dimension of time series Y, and the i^{th} column specifies the L-dimensional vector at index $\boldsymbol{x}_i$.

When the time series being analyzed has missing values, only a subset of the values in Y are observed. Therefore, given a set of observations $\boldsymbol{y}_o \subseteq \boldsymbol{y}$, the aim is to predict some of the unobserved values at some input locations for certain tasks (or variables). For this, L different processes (latent functions) are modeled, $\{f_l\}_{l=1}^{L}$, assuming that each l dimension is drawn from a f_l process.

The most straightforward way to model the L processes is to assume that they are independent and thus use a GP for each of those processes. That is, each process is defined by a mean function, $\mu_l(\mathbf{x})$, and a covariance function, $k_l(\mathbf{x}, \mathbf{x'})$. For convenience, we assume the mean function to be is zero. Then, $f_l(\boldsymbol{x}) \sim GP(0, k_l(\boldsymbol{x}, \boldsymbol{x}'))$, and

$$\boldsymbol{y}_l = \begin{bmatrix} y_{1l} \\ \vdots \\ y_{\tilde{T}_l l} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, K_l + \sigma_l^2 I), \quad \text{where } l \in \{1, ..., L\}$$

Additionally,

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_L \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & K_L \end{bmatrix} + \begin{bmatrix} \sigma_1^2 I & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \sigma_L^2 I \end{bmatrix} \right) = \mathcal{N}(\mathbf{0}, K_{f,f} + \Sigma_L) \tag{9}$$

where K_l is the covariance matrix associated with process f_l, and Σ_L is the $L \times L$ diagonal matrix in which the $(l,l)^{th}$ element is σ_l^2.

This approach assumes that the processes are independent, and thus, the blocks outside the main diagonal of $K_{f,f}$ are zero. Conversely, multi-task learning aims to exploit the dependencies between processes and define those terms outside the diagonal. In particular, this approach defines a covariance function that gives a positive semi-definite (PSD) covariance matrix $K_{f,f}$, also considering the dependencies between the processes.

Different models for defining the covariance function can be found in the literature. A widely used model is the Intrinsic Coregionalization Model (ICM) [36], which assumes that the $f_l(\boldsymbol{x})$ processes are defined by a linear combination of functions that have been sampled independently for the same GP, sharing the same covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$. That is,

$$f_l(\boldsymbol{x}) = \sum_{i=1}^{R} a_d^i u^i(\boldsymbol{x}) \tag{10}$$

where $\{f_l(\boldsymbol{x})\}_{l=1}^{L}$ is the set of functions to be modeled, $a_d^i \in \mathbb{R}$ are the coefficients of the linear combination, and each $u^i(\boldsymbol{x})$ is sampled from $u(\boldsymbol{x}) \sim GP(0, k(\boldsymbol{x}, \boldsymbol{x}'))$. Then, the covariance function is defined as

$$cov(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \boldsymbol{A}\boldsymbol{A}^T k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{B} k(\boldsymbol{x}, \boldsymbol{x}')$$

where $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), ..., f_L(\boldsymbol{x})]^T$, $\boldsymbol{A} = [\boldsymbol{a}^1 \quad \boldsymbol{a}^2 \quad ... \quad \boldsymbol{a}^R]$, $\boldsymbol{B} \in \mathbb{R}^{L \times L}$, and k is a covariance function over inputs. The main idea is to place independent GP priors over the processes, with a shared correlation function k over time.

Following this ICM model, [21] define the covariance function of the MGP as:

$$cov(f_{l_1}(\boldsymbol{x}), f_{l_2}(\boldsymbol{x}')) = K^f_{l_1,l_2} k(\boldsymbol{x}, \boldsymbol{x}')$$

where $y_{il} \sim N(f_l(\boldsymbol{x}_i), \sigma_l^2)$, $K^f \in \mathbb{R}^{L \times L}$ is a PSD matrix that specifies the inter-task similarities, and $K^f_{l_1,l_2}$ is the $(l_1, l_2)^{th}$ element of matrix K^f. That is,

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_L \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K^f_{11}K & \cdots & K^f_{1L}K \\ \vdots & \ddots & \vdots \\ K^f_{L1}K & \cdots & K^f_{LL}K \end{bmatrix} + \begin{bmatrix} \sigma_1^2 I & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \sigma_L^2 I \end{bmatrix} \right) = \mathcal{N}(\mathbf{0}, K^f \otimes K + \Sigma_L) \tag{11}$$

where $K_{f,f} = K^f \otimes K$.

Given the training index set X and the output observations $\boldsymbol{y}$, the posterior distribution of $\boldsymbol{f}(\boldsymbol{x}_*) = \{f_1(\boldsymbol{x}_*), ..., f_L(\boldsymbol{x}_*)\}$ at test point $\boldsymbol{x}_*$ is given by

$$\boldsymbol{f}(\boldsymbol{x}_*)|X, \boldsymbol{y}, \boldsymbol{x}_* \sim \mathcal{N}(\bar{\boldsymbol{f}}(\boldsymbol{x}_*), \Sigma_*) \tag{12}$$

where the mean and variance predictions are respectively given by

$$\begin{aligned} \bar{\boldsymbol{f}}(\boldsymbol{x}_*) =& (K^f \otimes K(\boldsymbol{x}_*, X))^T \Sigma^{-1} \boldsymbol{y} \\ \Sigma_* = var(\boldsymbol{x}_*) =& (K^f \otimes K(\boldsymbol{x}_*, \boldsymbol{x}_*)) - \\ & (K^f \otimes K(\boldsymbol{x}_*, X)) \Sigma^{-1} (K^f \otimes K(X, \boldsymbol{x}_*)) \end{aligned} \tag{13}$$

where $\otimes$ denotes the Kronecker product, $\Sigma = K^f \otimes K(X, X) + \Sigma_L \otimes I$ is a $L\tilde{T} \times L\tilde{T}$, K^f is the matrix that specifies the inter-task similarities, $K(X, X)$ is the matrix of covariances between all pairs of training points, Σ_L is the $L \times L$ diagonal matrix in which the $(l,l)^{th}$ element is σ_l^2,

and $K(\boldsymbol{x}_*, X)$ is the vector of covariances between the test point $\boldsymbol{x}_*$ and the training points.

Since only a subset of values $\boldsymbol{y}_o \subseteq \boldsymbol{y}$ has been observed, the covariance matrix Σ only needs to be computed at the observed values. That is, if the observed values $\boldsymbol{y}_o$ correspond to the values in the indexes I_o of the vector $\boldsymbol{y}$, then, from the matrix $(K^f \otimes K(\boldsymbol{x}_*, X))^T\Sigma^{-1}$ only the columns in those indices I_o are needed. This means that the covariance matrix Σ and its inverse only needs to be computed at the observed values. Additionally, from the matrix $(K^f \otimes K(\boldsymbol{x}_*, X))^T$, only the columns associated with the dimensions and time indexes with observations need to be computed (i.e., the columns in the I_o indexes).

Learning Hyperparameters

The parameters to be learned are $\boldsymbol{\theta} = (K^f, \{\sigma_l^2\}_{l=1}^L, \boldsymbol{\eta})$, where $\boldsymbol{\eta}$ are the parameters of the $k(\boldsymbol{x}, \boldsymbol{x}')$ kernel function. The aim is to learn the parameters $\boldsymbol{\theta}$ to maximize the marginal likelihood $p(\boldsymbol{y}_o|X, \boldsymbol{\theta})$. This can be done using 1) gradient-based methods, where the Cholesky decomposition can be used to guarantee positive-semidefiniteness of K^f (i.e., $K^f = LL^T$, where L is lower triangular), or 2) the EM algorithm.

Taking into account the fact that $\boldsymbol{y}|X \sim N(\boldsymbol{0}, \Sigma)$, the log marginal likelihood to be maximized is defined by:

$$\begin{aligned}\mathcal{L} &= log\ p(\boldsymbol{y}_o|X, \boldsymbol{\theta}) \\ &= -\frac{1}{2}log\ det(\Sigma_o) - \frac{1}{2}\boldsymbol{y}_o^T\Sigma_o^{-1}\boldsymbol{y}_o - \frac{n_o}{2}log(2\pi)\end{aligned} \tag{14}$$

where Σ_o is the covariance matrix at the observed values, and n_o is the length of vector $\boldsymbol{y}_o$.

REFERENCES

[1] T. Ruf. The Lomb-Scargle periodogram in biological rhythm research: Analysis of incomplete and unequally spaced time-series. *Biological Rhythm Research*, 30(2):178–201, 1999.

[2] J D Scargle. Studies in astronomical time seires analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853, 1982.

[3] H.M. Adorf. Interpolation of Irregularly Sampled Data Series - A Survey. *Astronomical Data Analysis Software and Systems IV*, 77:460, 1995.

[4] J . Belcher, J . S . Hampton, and G . Tunnicliffe Wilson. Parameterization of Continuous Time Autoregressive Models for Irregularly Sampled Time Series Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):141–155, 1994.

[5] Zachary Chase Lipton, David C. Kale, and Randall Wetzel. Modeling Missing Data in Clinical Time Series with RNNs. *Proceedings of Machine Learning for Healthcare*, 58(4):725–737, 2016.

[6] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):1–18, 2019.

[7] Yonghong Luo. Multivariate Time Series Imputation with Generative Adversarial Networks. In *NIPS*, number NeurIPS, 2018.

[8] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2GaN: End-to-end generative adversarial network for multivariate time series imputation. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-Augus:3094–3100, 2019.

[9] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1):1–12, 2018.

[10] Satya Narayan Shukla and Benjamin M. Marlin. Interpolation-prediction networks for irregularly sampled time series. *iclr*, (2017), 2019.

[11] Satya Narayan Shukla and Benjamin M. Marlin. Multi-Time Attention Networks for Irregularly Sampled Time Series. *ICLR*, pages 1–15, 2021.

[12] Joseph Futoma, Sanjay Hariharan, and Katherine Heller. Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier. *PMLR*, 2017.

[13] Joseph Futoma. Gaussian Process-Based Models for Clinical Time Series in Healthcare. *ProQuest Dissertations and Theses*, page 156, 2018.

[14] Steven Cheng Xian Li and Benjamin Marlin. A scalable end-to-end Gaussian process adapter for irregularly sampled time series classification. *Advances in Neural Information Processing Systems*, (Nips):1812–1820, 2016.

[15] Steven Cheng-xian Li and Benjamin Marlin. Classification of Sparse and Irregularly Sampled Time Series with Mixtures of Expected Gaussian Kernels and Random Features. 2015.

[16] Zitao Liu and Milos Hauskrecht. Learning adaptive forecasting models from irregularly sampled multivariate clinical data. In *30th AAAI Conference on Artificial Intelligence*, pages 1273–1279, 2016.

[17] Filippo Maria Bianchi, Lorenzo Livi, Karl Øyvind Mikalsen, Michael Kampffmeyer, and Robert Jenssen. Learning representations of multivariate time series with missing data. *Pattern Recognition*, 96:106973, 2019.

[18] P. Esling and C. Agon. Time-series data mining. *ACM Comput. Surv.*, 45(1):1–34, 2012.

[19] Benjamin M. Marlin, David C. Kale, Robinder G. Khemani, and Randall C. Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. *IHI'12 - Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 389–398, 2012.

[20] Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. Patient subtyping via time-aware LSTM networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1296:65–74, 2017.

[21] Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K.I. Williams. Multi-task Gaussian Process prediction. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 2009.

[22] Kalyanmoy Deb, Associate Member, Amrit Pratap, Sameer Agarwal, and T Meyarivan. A fast and elitist multi-objective genetic algorithm:NSGAII. 6(2):182–197, 2002.

[23] Bekele Afessa, Mark T. Keegan, Ognjen Gajic, Rolf D. Hubmayr, and Steve G. Peters. The influence of missing components of the Acute Physiology Score of APACHE

III on the measurement of ICU performance. *Intensive Care Medicine*, 31(11):1537–1543, 2005.

[24] Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):7576–7586, 2018.

[25] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 8:89497–89509, 2020.

[26] Karl Øyvind Mikalsen, Cristina Soguero-Ruiz, and Robert Jenssen. A Kernel to Exploit Informative Missingness in Multivariate Time Series from EHRs. *Studies in Computational Intelligence*, 914(9037):23–36, 2021.

[27] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. In *arXiv*, pages 1–36, 2018.

[28] Ary L Goldberger, Luis A N Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-kang Peng, and H Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation [Online].*, 101(23):e215–e220, 2000.

[29] Sakyajit Bhattacharya, Vaibhav Rajan, and Harsh Shrivastava. ICU mortality prediction: A classification algorithm for imbalanced datasets. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, number 1, pages 1288–1294, 2017.

[30] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153, 2013.

[31] Thach Le Nguyen, Severin Gsponer, Iulia Ilie, Martin O'Reilly, and Georgiana Ifrim. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. *Data Mining and Knowledge Discovery*, 33(4):1183–1222, 2019.

[32] Mohammad Shokoohi-Yekta, Bing Hu, Hongxia Jin, Jun Wang, and Eamonn Keogh. Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data Mining and Knowledge Discovery*, 31(1):1–31, 2017.

[33] Jason Lines, Sarah Taylor, and Anthony Bagnall. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5):1–35, 2018.

[34] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. sktime: A Unified Interface for Machine Learning with Time Series. In *arXiv*, pages 1–9, 2019.

[35] Kohei Hayashi, Takashi Takenouchi, Ryota Tomioka, and Hisashi Kashima. Self-measuring similarity for multi-task Gaussian process. In *JMLR: Workshop and Conference Proceedings*, volume 27, pages 103–110, 2012.

[36] H. Wackernagel. Multivariate Geostatistics: An Introduction with Applications. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, 8(33):363A, 1996.