Model selection to improve multiple imputation for handling high rate missingness in a water quality dataset

Romy Ratolojanahary, Raymond Houé Ngouna, Kamal Medjaher, Jean Junca-Bourié, Fabien Dauriac, Mathieu Sebilo

Model selection to improve multiple imputation for handling high rate missingness in a water quality dataset

Romy Ratolojanahary[a,*], Raymond Houé Ngouna[a], Kamal Medjaher[a], Jean Junca-Bourié[b], Fabien Dauriac[c], Mathieu Sebilo[d]

[a] *Laboratoire Génie de Production, École Nationale d'Ingénieurs de Tarbes, BP1629, 47 avenue d'Azereix, Tarbes Cedex 16 65016, France*
[b] *Agence de l'eau Adour-Garonne, Tarbes, 7 Passage de l'Europe, Pau 64000, France*
[c] *Chambre d'Agriculture des Hautes-Pyrénées, 20 Place du Foirail, Tarbes 65000, France*
[d] *IEES, Université Pierre et Marie Curie, 4 Place Jussieu, Paris 75005, France*

ARTICLE INFO

Keywords:
Multiple imputation
High missingness
Model selection
Machine learning
Data preprocessing
Water quality

ABSTRACT

In the current era of "information everywhere", extracting knowledge from a great amount of data is increasingly acknowledged as a promising channel for providing relevant insights to decision makers. One key issue encountered may be the poor quality of the raw data, particularly due to the high missingness, that may affect the quality and the relevance of the results' interpretation. Automating the exploration of the underlying data with powerful methods, allowing to handle missingness and then perform a learning process to discover relevant knowledge, can then be considered as a successful strategy for systems' monitoring. Within the context of water quality analysis, the aim of the present study is to propose a robust method for selecting the best algorithm to combine with MICE (Multivariate Imputations by Chained Equations) in order to handle multiple relationships between a high amount of features of interest (more than 200) concerned with a high rate of missingness (more than 80%). The main contribution is to improve MICE, taking advantage of the ability of Machine Learning algorithms to address complex relationships among a large number of parameters. The competing methods that are implemented are Random Forest (RF), Boosted Regression Trees (BRT), K- Nearest Neighbors (KNN) and Support Vector Regression (SVR). The obtained results show that the hybridization of MICE with SVR, KNN, RF and BRT performs better than the original MICE taken alone. Furthermore, MICE-SVR gives a good trade-off in terms of performance and computing time.

1. Introduction

The proliferation of sensing devices has increased the ability of organizations to acquire various and great amount of data, allowing them to implement real-time monitoring of their systems. This is generally based on the analyses of complex relationships between several factors of interest, such as in water quality analysis. Online monitoring has indeed offered the development of decision systems that are able to accelerate decision-making and anticipate actions to prevent undesired events or to eradicate critical issues. To achieve such a goal, it is required to pre-process the raw data, especially when some values are missing on a certain level.

Missing data is a recurring phenomenon in real-world applications (Sterne et al., 2009; Yang, Liu, Zeng, & Xie, 2019). It may occur due to sensor failures, bad or non-existing strategy for data acquisition, budget issues, lack of response from a participant in the case of survey or various other reasons. If the complete data are representative of the studied phenomenon, this missing information is negligible, otherwise the results may be incorrect and may lead to wrong interpretations. For example, anomalies could go undetected if they happen during a non-monitored period of time.

There are two ways of dealing with missing data: deletion or imputation (Buhi, 2008). Deletion means discarding the observations or the variables with missing data, which is called complete-case analysis, while imputation consists in reconstructing the missing values. Because of its simplicity, deletion is usually the default method used in practice. However, there are many cases in various fields in which this method showed some limitations. Indeed, it decreases the sample size and may lead to a loss of substantial information. In Clark and Altman (2003) for instance, the number of

* Corresponding author.
E-mail addresses: romy-alinoro.ratolojanahary@enit.fr (R. Ratolojanahary), raymond.houe-ngouna@enit.fr (R. Houé Ngouna), kamal.medjaher@enit.fr (K. Medjaher), jean.junca-bourie@eau-adour-garonne.fr (J. Junca-Bourié), fdauriac@hautes-pyrenees.chambagri.fr (F. Dauriac), mathieu.sebilo@upmc.fr (M. Sebilo).

observations dropped from 1189 to 518 (43% of the original data) in an ovarian cancer dataset, which led to biased interpretation.

Another deletion method is pairwise deletion through which only non-missing values are used for analyses, for instance in correlations scores calculation where the method fails when the two correlated variables are not filled at the same time. Instead of discarding an observation or a variable concerned with missing value, it is preferable to estimate accurately those missing values in order to provide relevant interpretations.

Quoting White and co-authors, "awareness has grown of the need to go beyond complete-analysis" and some major improvements of the simplistic methods have been proposed in the literature, since Rubin's innovative proposal for approaching missingness (White, Royston, & Wood, 2010). Among others, Rubin, who is the author of Multiple Imputation (MI), defined a conceptual framework for characterizing missing data that allows to distinguish various types and to determine when missing data can be ignored (Little & Rubin, 1987; Rubin, 1976). The major insight of the proposed imputation method is that it addresses uncertainty and complexity of the data structure, allowing to go beyond deleting or discarding data.

Following Rubin, van Buuren introduced the Multiple Imputations by Chained Equations (MICE), a MI technique that requires fewer assumptions on missingness and also handles relationships between variables (van Buuren & Groothuis-Oudshoorn, 2011). However, original MICE considers only linear relationships and has been successfully applied to dataset with at most 70% of missingness. It may therefore fail in other cases such as in water quality data as considered in the present study, which are characterized by a high rate of missingness and a great amount of factors of interest that are not necessarily linearly related. This suggests the need of an alternative method to improve the imputation mechanism in order to provide relevant interpretation of the results, which is the purpose of this work.

The rest of the paper is organized as follows: the main imputation methods available in the literature are reviewed in Section 2, followed by the presentation of a method to improve MICE for multiple data imputation in Section 3. An application of the proposed method on experimental dataset, along with associated results, are described in Section 4 while the last section contains the conclusion and perspectives of the present work.

2. Related work

In order to choose an appropriate method for handling missing data, the underlying cause of the missingness has to be investigated. Indeed, as mentionned in Buhi (2008), each method only works under certain assumptions, namely complete randomness, conditional randomness or systematic reasons.

2.1. Missingness patterns

The conceptual framework allowing to take into account certain assumptions, as noted above, has been defined by Rubin (1976). There are three types of missing data, depending on the missing mechanism : (1) Missing completely at Random (MCAR), (2) Missing at Random (MAR) and (3) Missing Not at Random (MNAR).

Let R be the locations of the missing data in a dataset $X = (X_{obs}, X_{miss})$, and ψ the parameters of the missing data model; where X_{obs} and X_{miss} are respectively the observed and the missing values. MCAR, MAR and MNAR patterns are formally defined as follows (van Buuren, 2018):

- Data are MCAR if the probability of missingness is independent of both the observed variables and the variables with missing values. This is the case, for example, when people forget to answer a question in a survey. Formally,

$$P(R = 0|X_{obs}, X_{miss}, \psi) = P(R = 0|\psi) \tag{1}$$

- Data are MAR if the probability of missingness is due entirely to the observed variables and is independent of the unseen data. In other words, the missingness is a function of some other observed variables in the dataset (for example, people of one sex are less likely to disclose their weight):

$$P(R = 0|X_{obs}, X_{miss}, \psi) = P(R = 0|X_{obs}, \psi) \tag{2}$$

  Therefore, MAR data are a good candidate for data imputation based on observed variables (Buhi, 2008).

- Data are MNAR if the missing value is related to the actual values (for example, people who weigh more are most likely to not disclose their weight):

$$P(R = 0|X_{obs}, X_{miss}, \psi) \tag{3}$$

  depends on all three elements.

When data are MNAR, the missingness process is called *non-ignorable*, meaning that the cause of the missingness must be included in the model, whereas MAR and MCAR data missingness processes are called *ignorable*. Following the assumptions behind these three patterns, several methods have been provided in the literature for solving appropriately the missingness.

2.2. Single imputation methods

Methods that compute one single value per missing data are referred as *single imputation* methods. The most common single imputation methods are mean, median or mode imputation, consisting in replacing the missing value with the mean, median or mode of the associated variable (Buhi, 2008). In this case, the missing value is easy to compute, but the method ignores the correlation among the variables and underestimates the standard deviation. If the variable containing missing values is categorical, a simple option is to create a new category for the missing values. This method is suitable for MNAR data, i.e. when the missingness is correlated to the values of the missing data. When a variable of the incomplete dataset is a periodic time series, a more elaborated single imputation technique is to apply a linear interpolation or an Autoregressive Integrated Moving Average (ARIMA) model to fill in the missing values (Shao, Meng, & Sun, 2016). Although those two techniques are simple, the first one is not efficient when the missing gap is large, and the second one requires a periodic time series. Another technique involves predicting the values from the observed variables. For example, K-nearest neighbors (KNN) replaces the missing value with a linear combination of the K nearest non-missing observations (Jordanov, Petrov, & Petrozziello, 2018; Tutz & Ramzan, 2015). To use this algorithm, it is necessary to choose the optimal K and define a distance measurement between two observations. A local similarity imputation based on Fast Clustering was proposed in Zhao, Chen, Yang, Hu, and Obaidat (2018). The authors partition the incomplete data with a fast clustering method (Stacked Autoencoder-based), then fill the missing data within each cluster using a KNN algorithm. The obtained results showed that the proposed method outperformed other local similarity-based methods. Shao and co-authors applied two Single Layer Feed Forward Neural Networks (Extreme Learning Machine and Radial Basis Function Network) on a periodic soil moisture time series (Shao et al., 2016). This method performed better predictions than a linear interpolation and ARIMA in infilling missing segments. However, it requires parameter tuning in order to be performing.

Table 1
Advantages and Drawbacks of the reported single imputation methods.

Method	Advantages	Drawbacks
Mean	Easy to implement	- Underestimates standard deviation - Ignores relationships between variables
Add a category	Easy to implement	Only works with categorical and MNAR data
Linear Interpolation	Takes time into account	Does not work when the missing gap is large
ARIMA	Takes time into account	Requires a periodic time series
Linear Regression	Takes into account relationships between variables	- Underestimates the variance - Ignores non linear relationships between variables
Stochastic linear regression	Takes into account relationships between variables	Ignores non linear relationships between variables
KNN	Takes into account relationships between variables	Requires parameter tuning
ANN	Takes into account the time factor	Requires parameter tuning

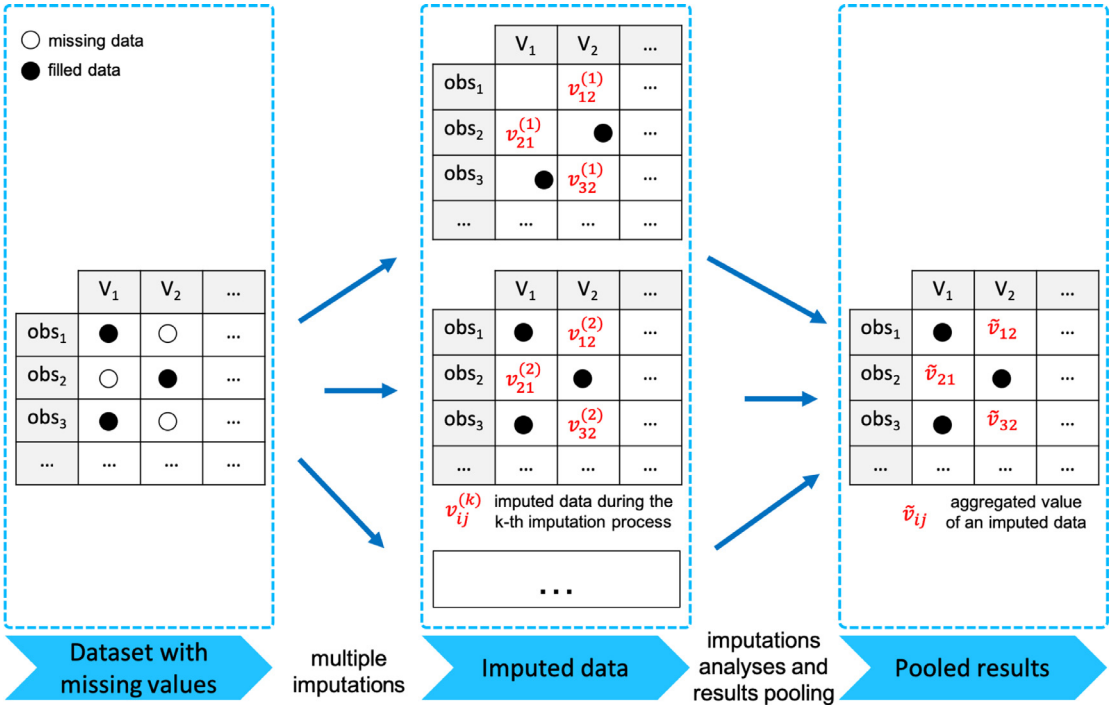


Fig. 1. Overview of the multiple imputation method.

A brief summary of these implementations of single imputation methods is presented in Table 1 that provides the main drawbacks and advantages. A well-known limitation that they have in common is that once a missing value is imputed, it is treated as a non-missing value.

2.3. Multiple imputation methods

In order to solve the limitations of single imputation, some authors have proposed to take into account the uncertainty of the imputed values (Little & Rubin, 1987; Neter, Maynes, & Ramanathan, 1965). In that purpose, Rubin has developed the Multiple Imputation (MI) method, which combines several single imputations (Little & Rubin, 1987), as described in the following.

2.3.1. Principles of multiple imputation

The principles of MI are illustrated in Fig. 1, based on the following main steps: (1) imputation phase where m datasets are produced by drawing them from a distribution, which can be different for each variable (van Buuren, 2018), (2) analysis phase in which the m datasets are analyzed, and (3) pooling phase that combines the m datasets to produce a final result, for example by calculating the mean of the imputed values for each missing value. The m datasets can be generated in parallel using parametric statistical theory and assuming a joint model for all the variables (van Buuren, 2007; Rubin & Schafer, 1990), such as in Multiple imputAtions of incoMplEte muLtIvariate dAta (AMELIA), which uses expectation-maximization with a bootstrapping algorithm (Honaker, King, & Blackwell, 2011). Such approach lacks flexibility and may lead to bias (van Buuren, 2007). The other alternative is to generate the m datasets until a stop criterion is met: in Hong and Wu (2011) for instance, the authors iteratively used association rules to successfully estimate the missing values. Although the studied dataset had a high missing rate, it was relatively small (there were only three variables). Some other examples of the sequential methods are Sequential Imputation for Missing Value (IMPSEQ) (Betrie, Sadiq, Tesfamariam, & Morin, 2014), a covariance-based imputation method and MICE, a series of linear regressions that consider a different distribution for each variable (van Buuren, 2007; Raghunathan, Lepkowski, Hoewyk, & Solenberger, 2001). Betrie and co-authors have found that the two sequential methods outperform AMELIA (Betrie et al., 2014). In Stekhoven and Buhlmann (2011), the authors introduced a MI method called MissForest, which is similar to MICE, except that it uses Random Forest instead of Linear Regression in the imputation step. As MissForest yielded a better performance than MICE, that result is encouraging towards tweaking the MICE algorithm, which is the object of the present work. A brief summary of

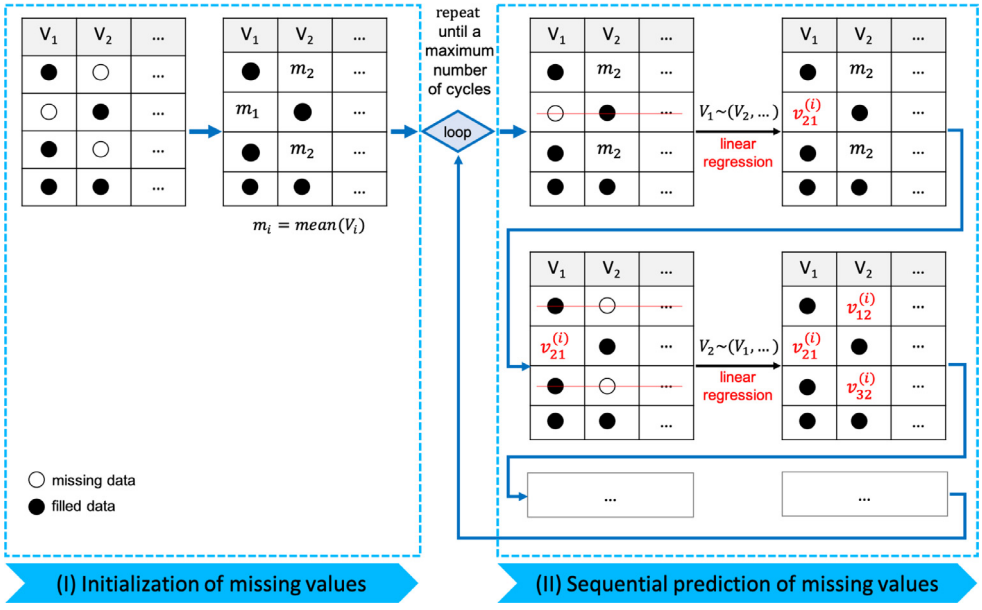


Fig. 2. Overview of the MICE algorithm.

Table 2
Advantages and Drawbacks of the reported MI methods.

Method	Advantages	Drawbacks
AMELIA	Can be applied to categorical, ordinal or continuous data	Assumes a joint model for all the variables
MI using decision rules	Works well when the missing-value rate is high	Not adapted to data with a large number of variables
IMPSEQ	Time complexity	- Lack of robustness toward outliers - Does not take into account nonlinear relationships between variables
MICE	Flexibility	- Does not take into account non-linear relationships between variables - Theoretical justification needed
MissForest	- Adapted to high dimensional datasets - Takes into account linear relationships between variables	Computation time issue

the advantages and drawbacks of the methods presented above is given in Table 2, while the original MICE principles are described in the following.

2.3.2. Main principles of MICE

The main steps of MICE are summarized in Fig. 2 and detailed in Algorithm 1. MICE algorithm implementation was based on a method described in Azur, Stuart, Frangakis, and Leaf (2011). It assumes that missing data are of MAR type. The first step is to initialize the missing values to the mean of each column. Then the missing values of the first variable are reset to "missing". After that, a regression model is fitted on the subset of the dataset where the value of this variable is present. Finally, the obtained model is used to fill in the value and update the dataset. This process is repeated for each variable until all the missing data are estimated. The whole process, first step excluded, is reiterated *n_cycles* times until the estimated data converge. In the literature, it is advised to increase the number of cycles in function of the size of the dataset and the missingness ratio (Graham, Olchowski, & Gilreath, 2007). Although MICE has been proved efficient in the literature, the trade-off between computational cost and performance becomes imbalanced when dealing with large datasets and/or datasets with a high missingness rate. Indeed, the number of imputed datasets has to be increased, and so does the computational time. Furthermore, a high missingness rate implies high uncertainty. Another key issue is that this form of the algorithm is based on linear regression, which may not reflect the actual relationships between the variables of the current study. To address these issues, an improved version of MICE is proposed and described in the following.

3. The proposed method to improve MICE

As noted above, the dataset concerned with water quality considered in this study has a very high missing rate (82%). Besides, there is a great amount of variables (more than 200) in which each is concerned with at least one missing value. The methods mentioned above, including the most performing, have been applied in a less constrained context and therefore, can fail to provide good results in the specific case of the dataset considered in this paper. It is then proposed to take advantage of the ability of Machine Learning algorithms for handling such issues in order to improve MICE. The two main ideas are: (1) define a set of competing methods, and then (2) replace the Linear Regression in the original MICE by each of these methods in order to select the most performing that fits the context of the present study.

The competing methods have been chosen among the most performing supervised learning algorithms in the literature, namely Random Forest(RF), Boosted Regression Trees (BRT), and Support Vector Regression (SVR). Besides, K-Nearest Neighbors (KNN), which is commonly used to solve missingness, has also been selected.

The main steps of the proposed method are illustrated in Fig. 3.

1. The first phase of the original MICE is initialized (step 1).
2. A competing method is then chosen, followed by a mechanism for optimally setting its hyperparameters (step 2).
3. Next, phase (II) of the original MICE is modified by replacing Linear Regression with the chosen method, and then launched in a loop that goes a number of times corresponding to the predefined number of cycles (step 3).

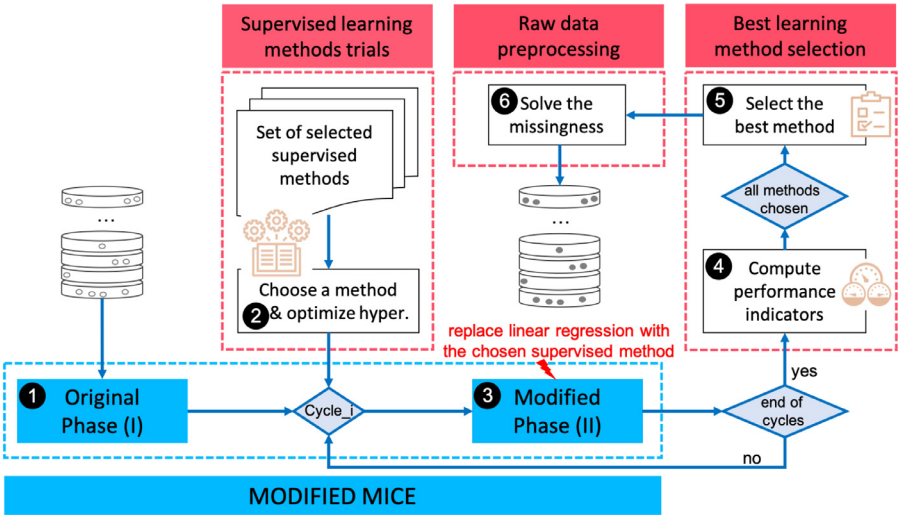


Fig. 3. The proposed method for model selection to improve MICE.

Algorithm 1 MICE.

Input:

- X incomplete data matrix of size $n\_obs \times n\_features$
- n_cycles number of cycles

Output:

- Completed data matrix of size $n\_obs - n\_features$

X_{full} := mean_impute(X)
for $i := 1$ to n_cycles **do**
  for $j := 1$ to $n\_features$ **do**
    $y_j := X_j$ /* the $j-th$ column */
    $X_{(j)} := X \setminus X_j$
    $m \subset \{1, n\} = \{i | X_j ! = NaN\}$ /* m denotes the indices where X_j is not missing */
    regressor := linear_regressor()
    regressor.fit($X^m_{(j)}, y^m_j$) /* the model is fitted on the subset of the dataset where X_j is not missing */
    $y^{\neg m}_{(j)}$:= regressor.predict($X^{\neg m}_{(j)}$) /* $\neg m$ denotes the indices where X_j is missing */
  end for
end for
return X_{full}

4. After convergence, performance indicators for the current method are computed (step 4).
5. When all the competing methods have been processed according to the four previous steps, a selection mechanism takes place by comparing their performance indicators (step 5).
6. Finally, the best method is applied to solve the missingness (step 6).

Due to the high missingness rate, the optimal choice of the hyperparameters (as considered in step 2) is based on a modified version of the studied dataset constructed according to the following procedure:

- For each variable, a triangular distribution is simulated with different parameters (min, mode, max). If a variable always has the same value, then that value is replicated in each observation. The triangular distribution has been used because it provides a simple representation of the real distribution of the dataset and allows more flexibility by taking into account the uncertainty of the values.
- The data are scaled so that the units of the variables do not play any role.
- The observations are shuffled and the missingness distribution of the real dataset is reproduced in order to mimic the real problem as accurately as possible.

Two main performance indicators have been used for the comparison (as realized in step 5), namely processing time and Mean Squared Error (MSE).

3.1. Theoretical background of the competing methods

3.1.1. Random Forest

Random Forest is an ensemble method based on fully grown regression trees. The objective is to build several weak learners (the regression trees) in parallel in order to produce a strong regressor. The main steps are as follows:

1. The observations are sampled with replacement (bootstrap aggregating).
2. A set of variables is selected randomly.
3. The tree is built upon the observations from step (1) and the variables from step (2).
4. The final prediction is made by averaging over the predictions of all decision trees.

In this algorithm, one of the most relevant hyperparameters to set in order to make the model perform well is the number of trees.

3.1.2. Boosted Regression Trees

Similarly to the Random Forest algorithm, BRT is an ensemble method based on regression trees. Gradient boosting is used to train the weak learners (shallow regression trees) sequentially. In this algorithm, a higher focus is set on observations that have higher errors on the previous tree and a gradient descent is used to minimize the loss function (least squared errors) at each step.

Let y_i be the target value and $f(x_i)$ its predictor.

The objective function is given as in Eq. (4):

$$L(y, f) = \sum_{i=1}^{n} l(y_i, f(x_i)) \tag{4}$$

where $l(y_i, f(x_i)) := (y_i - f(x_i))^2$.

The algoritm goes as follows:

f_0 is the trivial tree, it returns the mean value of Y.

For $k := 1$ to m:

- Calculate the negative gradient $-\Delta l(y_i, f(x_i))$, which corresponds to the residual for $i = 1$ to n.
- Fit a regression tree h_k for the residuals.
- Create a model $f_k = f_{k-1} + \nu\gamma_k h_k$, where γ is the step magnitude, found by searching $arg\min_\gamma \sum_{i=1}^{n} l(y_i, (f_{k-1}(x_i)) + \nu\gamma h_k(x_i)$, and ν is the learning rate.

Return f_m.

For this algorithm, the number of trees m, as well as the learning rate ν, are the hyperparameters that need to be set by the user in order for the method to perform well.

3.1.3. K-Nearest Neighbors

Let X and y be the training data, X^* a new observation and y^* the associated value to predict. The KNN algorithm goes through the following steps:

1. Calculate the distance between X^* and each of the observations of the training set;
2. Take the y values of the K closest observations $y_{i1}, y_{i2}, \ldots, y_{ik}$;
3. Assign to y^* a linear combination of these values (usually the mean).

Three hyperparameters have to be defined properly so that the algorithm performs well: the distance, the number of neighbors K and the type of aggregation of the neighbors values.

3.1.4. Support Vector Regression

Let X, y be a training data. The objective of SVR is to find a function f such that the deviation of $f(.)$ from the real values y is at most ε (Smola & Schölkopf, 2004). If the problem has no solution, slack variables ξ_i, ξ_i^* are introduced to tolerate part of the error. First, let's consider the case where f is linear, i.e. $f(x) = w.x + b$. f is then the solution of the following optimization problem (Eq. (5)):

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - < w, x_i > -b \leq \varepsilon + \xi_i \\ & < w, x_i > +b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{5}$$

where $C > 0$ is the trade-off between the flatness of f and the amount of tolerated deviations larger than ε, and $<,>$ is a scalar product. By using the dual representation of the problem based on Lagrange multipliers, we finally get: $f(x) = sum_{i=1}^{n}(\alpha_i + \alpha_i^*) < x_i, x > +b$ where α_i are the Lagrangian multipliers. If the adequate f is not linear, we can map the data into a high dimensional space where the function f becomes linear (Fig. 4). Instead of searching for the expression of ϕ, a function k called a kernel function, which satisfies $k(x, x) =< \phi(x), \phi(x) >$ is used. The existence of such a function is proved by the Mercer's theorem. ε, C and the kernel functions are the Support Vector Regression (SVR) hyperparameters that need to be selected properly for the performance of the algorithm.

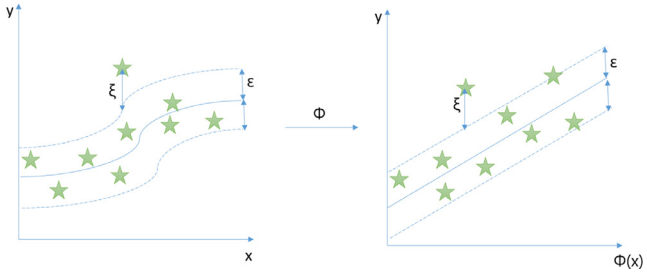


Fig. 4. Mapping to the feature space in SVR.

4. Application and results

4.1. The context of the study

The incomplete dataset used in this paper is taken from a water sample analysis made at Oursbelille, in the Adour plain, South-West of France, from 1991 to 2017. The operational principle of this drinking water collection point is described in Fig. 5. First, the water is pumped, its nitrate rate is measured and is conveyed to large aerial tanks in order to be treated by active charcoal. Then, the treated water is stored in a water tank. In a third step, some sensing devices are then used to monitor some quality indicators, such as the pH. In a fourth step, on demand, the stored water is chlorinated, before being dragged to another underground well, few kilometers away from the pumping well. From this second storage tank, water is distributed to the citizens of the Adour region. The region benefits of an oceanic climate, with a rainy winter and an average temperature ranging from 4 to 19 ◦ C.

The acquired data contain 148 observations of 411 water quality indicators, with an overall missingness of 82%. Fig. 6a is an overview of the dataset, where some of the measured water quality indicators are displayed, while Fig. 6b summarizes the missingness distribution per variable in the dataset.

Only the variables that are measured at least 5 times are considered, which reduces the dataset to 257 variables (52% of the 411 variables). It is noted that the removed variables do not restrict the analysis since they are not among the common hyperparameters for water quality assessment found in the literature.

4.2. Settings and assumptions of the implementation

Based on the presentation of the three missingness patterns, and the nature of the studied dataset (as described in the previous subsection), we can assume that our study is within the MAR pattern.

Moreover, the proposed method depends on several factors: (a) the number of cycles to perform the imputations, (b) the num-

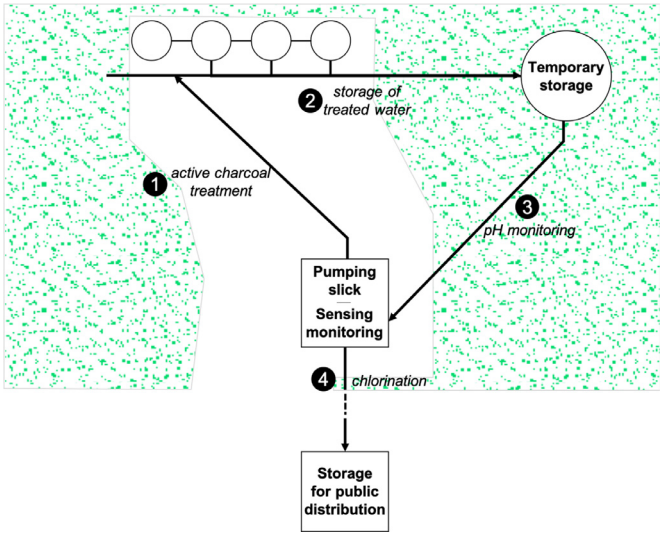
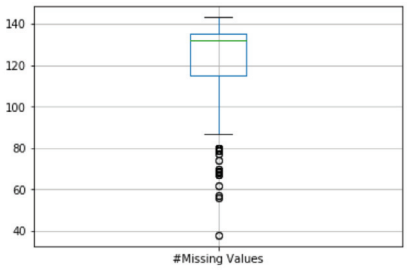


Fig. 5. Operational principle of the drinking water well of Oursbelille.

Date	Nitrates	Conductivity at 25°C	Total metolachlor	Simazine	Atrazine
1991-06-24	44.00	NaN	NaN	NaN	NaN
1992-06-23	45.00	NaN	NaN	NaN	NaN
1993-10-19	43.40	NaN	NaN	0.0	0.13
1994-08-23	43.00	NaN	NaN	NaN	NaN
1996-09-16	39.75	NaN	NaN	NaN	NaN

(a) Overview of the dataset.



(b) Number of missing values per variable.

Fig. 6. Description of the dataset.

ber of values defined for each hyperparameter, (c) the size of the dataset, (d) the number of variables of interest, and (e) the complexity of the ML algorithm itself. For these reasons, in order to obtain relevant results in a reasonable running time, and by opposition to what is commonly used in literature, only one value for the number of cycles (i.e. 10 cycles) is considered in this work.

The implementation of the proposed method was performed by using Python programming language, on a computer with the following main features:

- Operating System: Windows 10;
- RAM: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz;
- Processor: 8.00 Go.

The corresponding results are described and discussed in the following.

4.3. Implementation of the proposed method

The main steps of the proposed method have been implemented according to the following explanations.

- **Step 1.** The first phase of the original MICE, that is mean imputation, is launched (initialization step).
- **Step 2.** The next step concerns the hyperparameter tuning of the Machine Learning algorithms. There is no analytical solution that allows to find the optimal values. Therefore, to do so, a cross-validation is performed using the modified dataset, and a mean squared error (MSE) is measured. The optimal hyperparameters are therefore those that have the lowest MSE. Note that only a limited number of candidate values have been taken into account because adding more would drastically affect the algorithmic complexity.
  The competing methods are MICE, MICE combined with RF (MICE-RF), MICE combined with BRT (MICE-BRT), KNN (MICE-KNN) and MICE combined with SVR (MICE-SVR).
  The candidate values for the hyperparameters of the four Machine Learning algorithms (KNN, RF, BRT, SVR) are detailed in Table 3. For KNN, let us notice that since the studied dataset contains variables with only five non missing values, the number of neighbors is at most 4.
- **Step 3.** Phase (II) of the original MICE is modified by replacing Linear Regression with one of the competing algorithms, each with its optimal hyperparameters (as obtained in step 2).
- **Step 4.** The performance indicators, namely MSE and processing time, are computed for each algorithm.
- **Step 5.** The method that performed best in terms of MSE, and with a reasonable computing time, is then selected.
- **Step 6.** Finally, the winning method is used to solve the missingness.

Table 3
Candidate values of the hyperparameters for each Machine Learning method.

Algorithm	Hyperparameter	Candidate values
RF	n_trees	{10, 15, 20, 50, 100}
BRT	m	{30, 50, 100, 150}
	ν	{0.01, 0.1, 0.5}
KNN	K	{2, 3, 4}
	d	{euclidean, manhattan}
	y^*	{uniform, weighted}
SVR	ε	{0.01, 0.1}
	C	{0.01, 0.1, 1, 10, 100}
	kernel	{rbf, poly, sigmoid}
	γ	{1e-3, 0.01, 0.1, 1}

The main results of this implementation are presented and discussed in the next subsection.

4.4. Results and discussion

In the following, only steps 2, 4 and 5, which contain the main results of the implementation, are presented.

- **Step 2: Hyperparameter tuning.**

Random Forest. In this algorithm, the performance increases proportionally to the number of trees. However, it becomes rapidly time consuming. The objective is to find the smallest value for which the performance is good enough. Although it is not the optimal value, the number of trees is set to 15 in order to reduce the

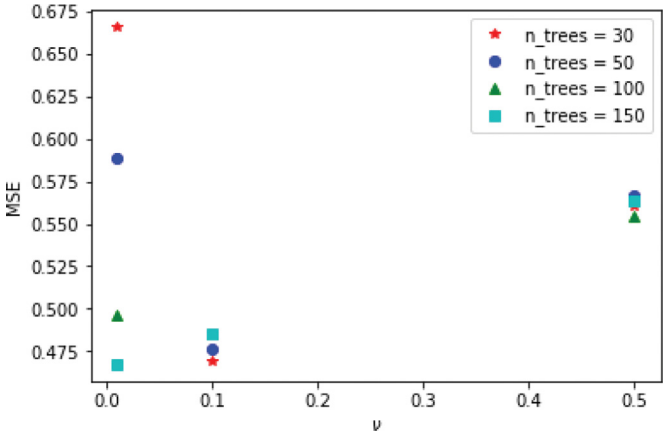
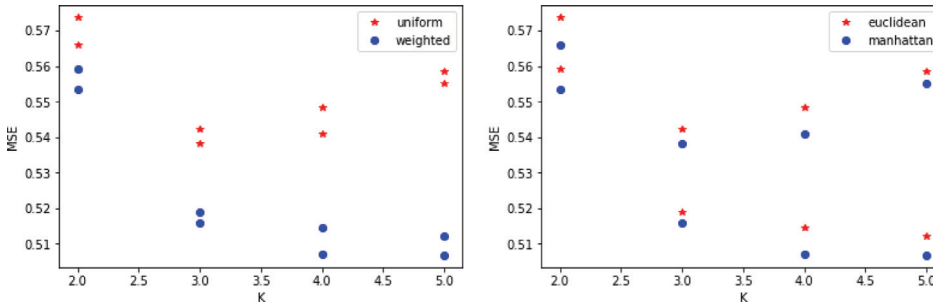
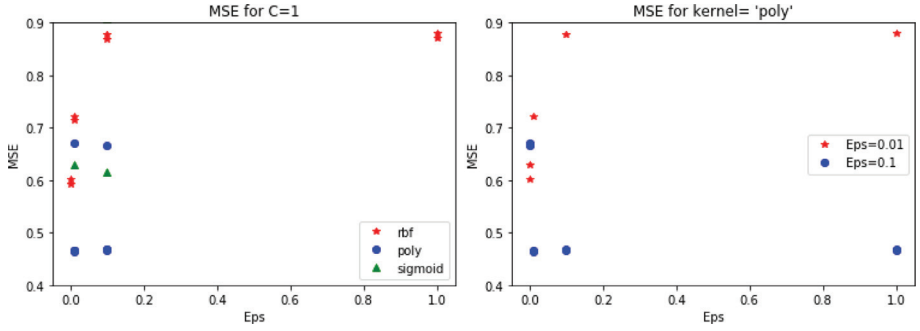


Fig. 7. Variation of MSE to choose the hyperparameters in BRT.



(a) Choice of K and the linear combination. (b) Choice of K and the distance.

Fig. 8. Variation of MSE to choose the hyperparameters in KNN.



(a) Choice of ε and the kernel function. (b) Choice of C and ε.

Fig. 9. Variation of MSE to choose the hyperparameters in SVR.

Table 4
Performance indicator (MSE) of the main RF hyperparameter.

n_trees	MSE
10	0.5159
15	0.4943
20	0.4850
50	0.4691
100	0.4653

computation time. Furthermore, the error does not decrease a lot between 15 and 100 estimators (see Table 4).

Boosted Regression Trees. Similarly to the previous algorithm, the best trade-off between computing time and performance is sought. It is noted that the number of trees is higher, because shallow trees are built in BRT instead of fully grown ones in RF.

Fig. 7 represents MSE in function of the learning rate ν, where the labels represent the number of trees. According to these results, the optimal hyperparameters for this study are $\nu = 0.01$ and $m = 150$. For computational time sake, hyperparameters with a slightly higher mean squared error (only a difference of 0.001) are chosen: $\nu = 0.1$ and $m = 30$.

K-Nearest Neighbors. For this algorithm, the hyperparameters to tune are the number of neighbors K, the distance d and the linear combination method of the neighbors value y^*. In this study, the euclidean distance is chosen, $K = 4$, and y^* is the weighted mean of the KNN. Their choice is illustrated in Fig. 8. Indeed, MSE score is lower for these values.

Support Vector Regression. For this algorithm, ε, C, the kernel function and the parameter γ associated to the kernel function need to be tuned. In Fig. 9, it is seen that the MSE is generally lowest for the polynomial kernel, and for $\varepsilon = 0.1$. The lowest MSE score is obtained with $\varepsilon = 0.01$, $C = 1$, $kernel = poly$ and the associated $\gamma = 0.01$.

Table 5
Performance indicator scores.

	MICE	MICE-SVR	MICE-4NN	MICE-RF	MICE-BRT
Processing time	6.87	**5.29**	8.25	65.18	32.59
MSE	1.09e24	**0.44**	0.58	0.55	0.54

- **Step 4: Computing the performance indicators**
  The results summarized in Table 5 show that MICE-SVR is the most performing method regarding both processing time (5.29 seconds) and MSE (0.44).
  The processing time was significantly high while combining MICE with RF and BRT. Indeed, all three methods, MICE, RF, and BRT are already computationally expensive by themselves. With a number of estimators set to 15 for Random Forest, a number of cycles set to 10 for MICE and 251 variables to fill, MICE-RF computes $15 \times 10 \times 251 = 37651$ fully grown regression trees. Similarly, MICE-BRT computes 43500 shallow regression trees.
  MICE performed the worst because in terms of MSE in the current implementation of the algorithm. Indeed, all the variables were used as predictors in the regression, whereas an intermediate variable selection step would have been appropriate. It also proves that the relationship between the variables are not linear.
  MICE-KNN is a little less performing than the other combinations of MICE with Machine Learning algorithms. This is due to the fact that the closest resembling observations are logically those that are closer in time. However, these values are not

systematically filled and the closest neighbors are only searched among non-missing observations for a given variable.

- **Step 5: Selection of the most performing method.** MICE-SVR performed best in both criteria, it is therefore the best performing competing method in this particular case.

The proposed methodology can handle datasets with a high missingness rate, and is also suitable for high-dimensional data. It is a flexible method that can take into account complex non-linear relationships between variables (if the competing methods are non-linear). It makes it possible to automate the selection of the best method to solve missingness, which reduces the amount of work of the data analyst, who can focus on tasks with higher added value, aiming at extracting knowledge. However, a few limitations are worth noting, particularly concerning the number of cycles preset to 10, and the relatively low number of potential hyperparameters values (that does not allow a rigorous sensitivity analysis of these hyperparameters). Furthermore, these parameters are tuned using an artificial dataset which has been constructed by modifying the real one. All these limitations are mainly due to algorithmic complexity, which constitutes by itself a challenge as well as a great scientific issue.

5. Conclusion and perspectives

It is widely acknowledged that data-driven methods provide powerful algorithms to analyze any issue that is of interest for decision-makers. However, performing such analyses with incomplete data may not be helpful to take reliable decisions. In this paper, a methodology for selecting the best algorithms to address the issue of data imputation, in the context of water quality assessment, has been proposed. A benchmark of four of the most powerful and commonly used ML algorithms has been performed for that purpose (Random Forest, Booted Regression Trees, K-Nearest Neighbors, Support Vector Regression). The results showed that MICE-SVR is the best in that it converges faster than the three others, and provides the best performance (notably in terms of prediction average error). It can then be applied to high missingness dataset, including data for water quality assessment that are often incomplete, as in the case of Adour (south-west of France) considered in the present study.

Based on the weaknesses of the proposed method, as mentioned in the discussion of the results, the following improvements are planned for further studies: (1) deeper automate the mechanism of the model selection by setting fuzzy rules in an inference engine that will aggregate all the performance indicators in a single indicator; (2) improve, for each competing method, the optimal choice of the hyperparameters using evolutionary algorithms in order to speed up the computing time and increase the number of values for each hyperparameter; (3) automate the choice of the number of cycles needed for the convergence of the imputations by taking into account the size of the data and its missingness rate; (4) introduce the temporal dimension within the imputation process.

Conflict of interest

There is no conflict of interest.

Credit authorship contribution statement

Romy Ratolojanahary: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft. **Raymond Houé Ngouna:** Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing, Supervision. **Kamal Medjaher:** Conceptualization, Methodology, Investigation, Writing - original draft, Writing - review & editing, Supervision, Funding acquisition. **Jean Junca-Bourié:** Investigation, Resources, Supervision, Funding acquisition. **Fabien Dauriac:** Investigation, Resources, Funding acquisition. **Mathieu Sebilo:** Investigation, Writing - review & editing, Supervision.

References

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research, 20*(1), 40–49. doi:10.1002/mpr.329.

Betrie, G. D., Sadiq, R., Tesfamariam, S., & Morin, K. A. (2014). On the issue of incomplete and missing water-quality data in mine site databases: Comparing three imputation methods. *Mine Water and the Environment, 35*(1), 3–9. doi:10.1007/s10230-014-0322-4.

Buhi, E. (2008). Out of sight, not out of mind: Strategies for handling missing data. *American Journal of Health Behavior, 32*(1). doi:10.5993/ajhb.32.1.8.

van Buuren, S. (2007). Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research, 16*(3), 219–242. doi:10.1177/0962280206074463.

van Buuren, S. (2018). *Flexible imputation of missing data*. Chapman and Hall/CRC.

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). Mice: Multivariate imputation by chained equations inr. *Journal of Statistical Software, 45*(3). doi:10.18637/jss.v045.i03.

Clark, T. G., & Altman, D. G. (2003). Developing a prognostic model in the presence of missing data. *Journal of Clinical Epidemiology, 56*(1), 28–37. doi:10.1016/s0895-4356(02)00539-5.

Graham, J. W., Olchowski, A. E., & Gilreath, T. D. (2007). How many imputations are really needed? some practical clarifications of multiple imputation theory. *Prevention Science, 8*(3), 206–213. doi:10.1007/s11121-007-0070-9.

Honaker, J., King, G., & Blackwell, M. (2011). Ameliaii: A program for missing data. *Journal of Statistical Software, 45*(7). doi:10.18637/jss.v045.i07.

Hong, T.-P., & Wu, C.-W. (2011). Mining rules from an incomplete dataset with a high missing rate. *Expert Systems with Applications, 38*(4), 3931–3936. doi:10.1016/j.eswa.2010.09.054.

Jordanov, I., Petrov, N., & Petrozziello, A. (2018). Classifiers accuracy improvement based on missing data imputation. *Journal of Artificial Intelligence and Soft Computing Research, 8*(1). doi:10.1515/jaiscr-2018-0002.

Little, R. J. A., & Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, Inc.. doi:10.1002/9780470316696.

Neter, J., Maynes, E. S., & Ramanathan, R. (1965). The effect of mismatching on the measurement of response errors. *Journal of the American Statistical Association, 60*(312), 1005–1027. doi:10.1080/01621459.1965.10480846.

Raghunathan, T. E., Lepkowski, J. M., Hoewyk, J. V., & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology, 27*(1), 85–95.

Rubin, D. B. (1976). Inference and missing data. *Biometrika, 63*(3), 581–592. doi:10.1093/biomet/63.3.581.

Rubin, D. B., & Schafer, J. L. (1990). Efficiently creating multiple imputations for incomplete multivariate normal data. In *Proceedings of the statistical computing section of the American statistical association*.

Shao, J., Meng, W., & Sun, G. (2016). Evaluation of missing value imputation methods for wireless soil datasets. *Personal and Ubiquitous Computing, 21*(1), 113–123. doi:10.1007/s00779-016-0978-9.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222. doi:10.1023/b:stco.0000035301.49549.88.

Stekhoven, D. J., & Buhlmann, P. (2011). MissForest–non-parametric missing value imputation for mixed-type data. *Bioinformatics, 28*(1), 112–118. doi:10.1093/bioinformatics/btr597.

Sterne, J. A. C., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., et al. (2009). Multiple imputation for missing data in epidemiological and clinical research: Potential and pitfalls. *BMJ, 338*(jun29 1). doi:10.1136/bmj.b2393. b2393–b2393

Tutz, G., & Ramzan, S. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics & Data Analysis, 90*, 84–99. doi:10.1016/j.csda.2015.04.009.

White, I. R., Royston, P., & Wood, A. M. (2010). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine, 30*(4), 377–399. doi:10.1002/sim.4067.

Yang, C., Liu, J., Zeng, Y., & Xie, G. (2019). Real-time condition monitoring and fault detection of components based on machine-learning reconstruction model. *Renewable Energy, 133*, 433–441. doi:10.1016/j.renene.2018.10.062.

Zhao, L., Chen, Z., Yang, Z., Hu, Y., & Obaidat, M. S. (2018). Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems. *IEEE Systems Journal, 12*(2), 1610–1620. doi:10.1109/jsyst.2016.2576026.