

# **Real-time data-driven missing data imputation for short-term sensor data of marine systems. A comparative study**

Christian Velasco-Gallego<sup>a, \*</sup>, Iraklis Lazakis<sup>a</sup>

<sup>a</sup>Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, 100 Montrose Street, G4 0LZ, Glasgow, United Kingdom

## **Abstract**

In the maritime industry, sensors are utilised to implement condition-based maintenance (CBM) to assist decision-making processes for energy efficient operations of marine machinery. However, the employment of sensors presents several challenges including the imputation of missing values. Data imputation is a crucial pre-processing step, the aim of which is the estimation of identified missing values to avoid under-utilisation of data that can lead to biased results. Although various studies have been developed on this topic, none of the studies so far have considered the option of imputing incomplete values in real-time to assist instant data-driven decision-making strategies. Hence, a methodological comparative study has been developed that examines a total of 20 widely implemented machine learning and time series forecasting algorithms. Moreover, a case study on a total of 7 machinery system parameters obtained from sensors installed on a cargo vessel is utilised to highlight the implementation of the proposed methodology. To assess the models' performance seven metrics are estimated (Execution time, MSE, MSLE, RMSE, MAPE, MedAE, Max Error). In all cases, ARIMA outperforms the remaining models, yielding a MedAE of 0.08 r/min and a Max Error of 2.4 r/min regarding the main engine rotational speed parameter.

**Keywords.** Data imputation, Machine learning, Time series forecasting, Marine machinery systems, condition-based maintenance (CBM), energy efficient operations.

---

\* Corresponding author

Email address: [christian.velasco@strath.ac.uk](mailto:christian.velasco@strath.ac.uk) (Christian Velasco-Gallego)

## Nomenclature

---

|         |   |
|---------|---|
| ARIMA   | Autoregressive Integrated Moving Average        |
| bPCA    | Bayesian Principal Component Analysis           |
| CBM     | Condition-Based Maintenance                     |
| CM      | Condition Monitoring                            |
| DNN     | Deep Neural Network                             |
| DTR     | Decision Tree Regressor                         |
| EM      | Expectation-Maximization                        |
| GAN     | Generative Adversarial Network                  |
| GHG     | Greenhouse Gas                                  |
| IIoT    | Industrial Internet of Things                   |
| IMO     | International Maritime Organization             |
| $k$ -NN | $k$ -Nearest Neighbor                           |
| LASSO   | Least Absolute Shrinkage and Selection Operator |
| MAE     | Mean Absolute Error                             |
| MAPE    | Mean Absolute Percentage Error                  |
| MAR     | Missing at Random                               |
| MARSS   | Multivariate Auto-Regressive State Space        |
| MCAR    | Missing Completely at Random                    |
| MCL     | Context and Linear Mean                         |
| MedAE   | Median Absolute Error                           |
| MEPC    | Marine Environment Protection Committee         |
| MI      | Multiple Imputation                             |
| MICE    | Multiple Imputation by Chained Equations        |
| ML      | Machine Learning                                |
| MLE     | Maximum Likelihood Estimation                   |
| MSE     | Mean Squared Error                              |
| NN      | Neural Network                                  |
| NRMSD   | Normalized Root Mean Squared Difference         |
| OEMs    | Original Equipment Manufacturers                |
| OLS     | Ordinary Least Squares                          |
| PEM     | Prediction Error Minimization                   |
| PLS     | Partial Least Squares                           |
| PMF     | Probability Matrix Factorization                |
| PPCA    | Probabilistic Principal Component Analysis      |
| PSF     | Pattern Sequence based Forecasting              |
| $R^2$   | Coefficient of determination                    |
| RBF     | Radial Basis Function                           |
| RMSE    | Root Mean Square Error                          |
| SMAPE   | Symmetric Mean Absolute Percentage Error        |
| STL     | Seasonal and Trend decomposition using Loess    |
| SVM     | Support Vector Machine                          |
| VAR     | Vector Autoregression                           |

---

## 1. Introduction

According to the Third International Maritime Organization (IMO) Greenhouse Gas (GHG) Study 2014,  
5 international shipping emitted 796 million tonnes of CO<sub>2</sub> in 2012, which constituted approximately 2.2% of the global anthropogenic CO<sub>2</sub> emissions in that year. These emissions, if not tackled, are expected to increase between 50% and 250% by 2050 as a result of the world maritime trade growth. Therefore, IMO's Marine Environment Protection

Committee (MEPC) considers extensively the control of GHG emissions from ships by adopting advanced regulations. An example of which is the approval of a roadmap to develop a comprehensive IMO strategy on reducing GHG emissions from ships, which includes the collection of data on ship's fuel oil consumption (IMO, 2020). Hence, the utilisation of sensors needs to be implemented, which implies the appliance of advanced monitoring techniques that not only provide an enhancement of the vessel efficiency, and thus reducing fuel oil consumption, but also aim to reduce the number of failures associated with machinery. On this basis, it is crucial to ensure the proper functioning of systems by establishing maintenance and inspection strategies.

The maritime industry currently offers state-of-the-art maintenance and inspection processes, an example of which is Condition-Based Maintenance (CBM). This is a maintenance strategy based on the condition monitoring of assets in order to reduce the number of failures associated with machinery (Lazakis et al., 2016). Condition Monitoring (CM) has proven to increase safety and reduce risk. Furthermore, if adequately implemented, CM could also increase the efficiency, reliability, profitability, and performance of the vessel, and thus facilitate the emissions reduction during its operational lifetime (Cheliotis et al., 2019).

Owing to these advantages, a large number of sensors are installed alongside the most critical components and around the environment where these assets are operating in order to implement CM effectively by utilising Industrial Internet of Things (IIoT). By employing IIoT, real-time data collection can be performed with the utilisation of smart sensors, reliable communications, and seamless integration, enabling predictive maintenance by the provision of relevant information (Aheleroff et al., 2020). Thus, diagnosis and prognosis can be performed to assess the current and future health of machinery to assist the decision-making processes, and then optimise, among other aspects, maintenance and inspection tasks, crew management, and spare parts stocks.

However, despite the undeniable benefits of the implementation of IIoT in the maritime industry, several challenges need to be tackled due to the employment of this technology. These include unreliable outcomes caused by certain anomalies and missing values that are originated by device failure, network collapse, and human error (Balakrishnan and Sangaiah, 2018; Izonin et al., 2019; Noor et al. 2014). Consequently, if missing values are not treated, the results derived from data analysis may be unreliable and inaccurate, leading to bias in further steps, and thus obtaining poor models used in decision-making processes (Fekade et al., 2018). Accordingly, data imputation is considered a crucial step in sensor data pre-processing as it deals with incomplete data that continues to be a challenging problem (Liu et al., 2020; Bashir et al., 2018).

Several studies have been performed to propose new data imputation methods in various fields, such as health and manufacturing industries, where the utilisation of sensors is highly expanded. These studies concluded the importance of implementing modern techniques to impute incomplete values, as traditional methods, such as mean imputation and deletion methods, may lead to bias in the estimates due to their lack of accuracy, and thus resulting in

40 a decrease of data quality (Azimi et al., 2019). Conversely, the utilisation of modern techniques, such as Multiple Imputation (MI), leads to accurate results and considers statistical uncertainty by adding an error term in the regression equations, and thus increasing the quality of the source sensor data with incomplete values (Hegde et al., 2019).

Nevertheless, the study of data imputation methods in the maritime industry is still inconsistent, as only a few approaches for imputing incomplete values have been conducted. Hence, the aim of this inquiry is to perform a comparative study of data imputation methods based on machine learning and time series forecasting models that are currently being implemented in other industries successfully.

The comparative study is subdivided into univariate and multivariate data imputation methods. Univariate methods impute values of a feature by only considering the feature being analysed. In general terms, the univariate methods analysed are sectioned in mean imputation, time series decomposition techniques, exponential smoothing methods, and Autoregressive Integrated Moving Average (ARIMA) models. By contrast, multivariate methods impute values of a feature by considering other features that are correlated with the feature being analysed. These methods are sectioned in linear regression,  $k$ -Nearest Neighbors ( $k$ -NN), Support Vector Machines (SVMs) for regression, Neural Networks (NNs), Vector Autoregressions (VARs), Decision Tree Regressors (DTRs), and ensemble methods.

The following paragraphs are structured as follows. Section 2 presents analogous works on data imputation methods. Section 3 describes the proposed methodology. Section 4 reflects on the results obtained after implementing the proposed methodology through a case study. Lastly, in Section 5 the conclusions are presented.

## 2. Literature review

Pratama et al. (2016) performed a review study of conventional imputation methods (ignoring, deletion, and mean/mode imputation) and of more modern imputation procedures (hot and cold deck imputation, and multiple imputation that included autoregressive models, genetic algorithm optimisation based methods, Support Vector Machines (SVMs), interpolation, maximum likelihood, fussy-rough set, and similarity measurements imputation methods). The study concluded that genetic algorithms, fuzzy c-means, and autoregressive methods were considered the best in terms of flexibility among the other imputation methods that were analysed.

Chong et al. (2016) implemented a comparative study of five imputation methods (linear regression, weighted  $k$ -NN, SVM, mean imputation, and replacing incomplete values with zero). The different models were evaluated by using time-series data collected from sensors installed in a floor of a community centre. To study how the performance of the analysed methods were affected by the ratio of missing values within the sample, a total of four datasets with different percentage of incomplete values were evaluated (5%, 10%, 15%, and 20%) by estimating the Normalized Root Mean Squared Difference (NRMSD). Linear regression,  $k$ -NN, and SVM provided more accurate results than mean imputation or replacing incomplete values with zero. It was also concluded that linear regression yielded better

accuracy when there was a linear relationship between the outcome and the predictors. Thus, the study suggested implementing SVM when the relationship between features is non-linear.

75 [Noor et al. \(2014\)](#) implemented two data imputation methods (linear interpolation and mean imputation) to assess their imputation accuracy by evaluating five randomly simulated missing data patterns divided into three degrees of complexity (small percentages of incomplete values (5%), medium percentages of incomplete values (15% and 25%), and large percentages of incomplete values (40%)). To assess their performance three metrics were estimated (Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination ( $R^2$ )). The results demonstrated that linear interpolation presented more accurate results in all simulations, as imputing  
80 incomplete values with the mean may distort the distribution of the feature and the relationships between variables may result in a degradation of the performance.

[Balakirshnan and Sangaiah \(2018\)](#) proposed an automated framework to impute missing values by applying the Context and Linear Mean (MCL) method. A temperature sensor was utilised to implement the MCL model in order to impute missing data. The performance of the model was evaluated with the response of the event when an  
85 incomplete value needed to be interpreted. Results presented a slight performance enhancement when applying the MCL model to impute incomplete values.

[Azimi et al. \(2019\)](#) proposed a multiple imputation approach to impute incomplete values to deliver decisions during the entire monitoring time. The proposed approach considered data variability by using context information, as the methods applied were selected based on the characteristics of the data and the auxiliary information type. Firstly,  
90 short-term history of data was utilised to impute values, as it is strongly correlated with the missing values when both the individual condition and the context situation are constant. Thus, autoregressive models were implemented when dealing with short-term data. Context data and lifestyle data were also utilised to impute missing values. To evaluate the performance of the model a case study was performed where 20 pregnant women were remotely monitored for seven months. A comparative study was implemented to evaluate the performance of the proposed method in  
95 comparison with four existing data imputation techniques ( $k$ -NN, Autoregressive, Maximum Likelihood Estimation (MLE) (Logistic), and SVM) by estimating RMSE and C-Index. It was concluded that when the incomplete window was small the proposed method led to less accurate results than autoregressive and  $k$ -NN models. Conversely, it obtained the best results when the incomplete window was large. Furthermore, it was also concluded that the bias of the estimates was minimised when a high correlation existed between context and incomplete values.

100 [Priya Stella Mary and Arockiam \(2017\)](#) introduced a methodology based on the assumption that data collected from sensors presented a highly spatial and temporal correlation. Thus, sensors that were close geographically presented a strong relationship during a certain amount of time. The proposed methodology estimated  $n$  proximate sensors by using geographical coordinates through the Haversine formula. It was then calculated if the

sensor data with incomplete values presented a linear relationship with the estimated sensor data obtained in the previous step. If a strong correlation between variables was identified, the incomplete values were imputed by utilising the correlated sensor occurrences at the corresponding time. To assess the performance of the proposed methodology a comparative study was performed with four existing imputation techniques (mean imputation, median imputation, mode imputation, and Multiple Imputation by Chained Equations (MICE)) and the RMSE was estimated to quantify their accuracy. Results indicated that the proposed model led to higher accuracy only when the relationship between the sensors was strong. Hence, it was not recommended to implement this approach when the sensors were not correlated either geographically or temporarily, as the accuracy of the imputation could decrease significantly. [Fekade et al. \(2018\)](#) presented a similar approach by proposing a probabilistic method to impute missing values from IoT devices by utilising data from analogous sensors. To divide the sensors to identify which IoT devices are neighbours, the  $k$ -means clustering algorithm was implemented. Once the different clusters were formed, a Probabilistic Matrix Factorization (PMF) was utilised inside each partition to impute missing values. The proposed methodology was compared with two existing algorithms (SVM with linear and Radial Basis Function (RBF) kernels, and Deep Neural Networks (DNNs) with two and three hidden layers) by estimating RMSE. Results indicated that the proposed methodologies were more efficient than SVM and DNN in terms of accuracy, as the existing methods were basically designed for classification purposes. Furthermore, it was also identified that the proposed method presented better results when the number of clusters was higher, as clusters with smaller groups would propitiate lesser differences between sensor measurements that belonged to the same group.

[Bashir and Wei \(2018\)](#) developed a new algorithm by the utilization of VAR to handle missing data by combining the Prediction Error Minimization (PEM) with an Expectation-Maximization (EM) algorithm, naming this new method as Vector Autoregressive Imputation Method (VAR-IM). In the first step of VAR-IM, linear interpolation was performed to provide a first estimation of missing values. Subsequently, a VAR( $p$ ) model was implemented to select the best lag value  $p$ . Lastly, EM and PME algorithms were applied alternately to estimate the best parameters of the VAR( $p$ ) model. 10% and 20% Missing Completely at Random (MCAR) datasets were created to evaluate the performance of the new algorithm alongside a comparative study where other five existing methods (listwise deletion, linear regression imputation, Multivariate Auto-Regressive State Space (MARSS) model, and EM algorithm) were also analysed. In all cases, the proposed method VAR-IM yielded the best results. However, the proposed algorithm presented some limitations, such as the necessity of the data to be stationary. Furthermore, the proposed methodology did not present more accurate results than the analysed techniques when the percentage of missing values was relatively small (approximately 5%).

[Izonin et al. \(2019\)](#) established another data imputation method based on the use of the Ito decomposition and the AdaBoost algorithm. To evaluate its performance several indicators were calculated (MAPE, RMSE, MAE,

and SMAPE). The developed regressor led to more accurate results than other algorithms that were also implemented, such as SVR and SGD regressor, although the proposed approach yielded lesser performance if either the data presented anomalies or the sample size was not adequately large to train the model.

[Liu et al. \(2020\)](#) proposed a univariate data imputation method to recover large gaps of missing data within the dataset. It utilised STL decomposition imputation, which tried to predict the incomplete values by implementing pattern discovery, and thus decomposing the time series into trend, seasonal, and remainder components. Hence, the repeated patterns could be learned from the time series, and then the imputation results were obtained by combining the estimated components of each gap. As STL decomposition required a complete dataset, a linear interpolation imputation was implemented in the first place. The proposed algorithm, named Itr-MS-STLDecImp, outperformed the other analysed methods when dealing with large gaps, although its accuracy decreased when the time series did not present trend and seasonality. [Bokde et al. \(2018\)](#) introduced another univariate data imputation method named imputePSF, which was an adjustment of the Pattern Sequence based Forecasting (PSF) algorithm that incorporated control structures to assess characteristics that were repeated along the time-series data utilised to impute values in a statistical basis. Firstly, the algorithm implemented the *k*-means algorithm to cluster the time series into different partitions, and then the resulting clusters were utilised as data input for the PSF model. The proposed approach outperformed the other imputation methods analysed, such as random forest methods and bayesian Principal Component Analysis (bPCA), when the time series presented periodic components. Thus, imputePSF was not recommended when the time series presented noisy trends or non-cyclical patterns.

[Hegde et al. \(2019\)](#) tackled a comparison of Probabilistic Principal Component Analysis (PPCA) and Multiple Imputation using Chained Equations (MICE) by evaluating 116 dental variables where incomplete values were generated at random. PCA was used for dimensionality reduction to obtain a lower dimensional space of the dataset. This property was utilised to impute the incomplete values, as the missing values were recovered from the compressed information distribution estimated by the PCA method. Then, EM algorithm was applied to estimate the MLE of an incomplete dataset in an iterative manner. Instead, MICE imputed the incomplete values multiple times by using regression models, and thus contemplating the statistical uncertainty in the imputations. For this study, logistic regression was utilised for nominal categorical variables (2 levels), polytomous logistic regression for nominal categorical variables (>2 levels), and predictive mean matching for continuous variables. RMSE was calculated to evaluate the performances of PPCA and MICE, which indicated that PPCA led to more efficient results in comparison with MICE.

[Hadeed et al. \(2020\)](#) implemented an evaluation process to assess existing imputation methods. The assessed imputation methods were divided into univariate imputation techniques (Mean, Median, Last Observation Carried Forward, Kalman Filter, Random, Markov) and multivariate time series (Predictive Mean Matching, Row Mean

Method). The performance of above methods was assessed by the implementation of five error metrics (Absolute Bias, Percent Absolute Error in Means,  $R^2$  Coefficient of Determination, Root Mean Square Error, Mean Absolute Error). For this analysis, a total of 20 household with complete 24-hour monitoring data for  $PM_{2.5}$  were utilised. The results suggested the Markov technique as the most promising approach. Kalman Filter also performed exceptionally well in data with strong trends. The Random technique also performed exceptionally well at high and low levels of missingness due to the number of iterations used to generate 1-minute concentration that were utilised to impute missing values within households. However, multivariate imputation methods presented a lower performance due to the significant differences between households.

[Chivers et al. \(2020\)](#) developed a two-step approach combining a binary classification step to identify the unbalance samples of rain and no-rain, and subsequently, apply regression analysis so that the magnitude of rain samples could be quantified when they occurred. The analysis encompassed the utilisation and comparison of commonly machine learning techniques, which included gradient boosting, bagged decision trees, neural networks, and support vector machines. This recovery approach of missing precipitation data was implemented through a case study of a network of weather stations and a network of rain gauges in England, UK, in data from a temperate oceanic climate at sub-hourly temporal resolution. The developed technique outperformed a surface fitting technique for the recovery of missing precipitation data at 30-minute resolution.

In the maritime industry, [Cheliotis et al. \(2019\)](#) developed a hybrid imputation method combining  $k$ -NN and MICE algorithms with first-principle knowledge. The proposed hybrid imputation method was compared with  $k$ -NN and MICE algorithms by estimating APE, MAPE, and the standard deviation of the error. To this end, all three methods were applied to time-series data collected from a total of 8 sensors coupled to the turbocharger and to the main engine of a chemical tanker. Results demonstrated that the proposed hybrid imputation model outperformed  $k$ -NN and MICE methods.

[Beck M. W. et. al \(2018\)](#) implemented a comparative methodology package in R named `imputeTestbench` to assess the prediction accuracy of different methods as imputation approaches for univariate time series. The proposed methodology tackled several challenges that need to be addressed when dealing with missing values, such as the amount of missing values, the accuracy of the imputation methods when dealing with missing values that are either Missing Completely at Random (MCAR) or Missing at Random (MAR), and the influence of the error metric chosen to assess the imputation performance.

Several deep learning methodologies were analysed to assess if their implementation could outperform classical imputation approaches. [Fortuin et al. \(2020\)](#) introduced an architecture based on a deep probabilistic model for multivariate time series imputation. The proposed methodology implemented deep variational autoencoders to map the missing values into a latent space without missingness, in which the low-dimensional dynamics were

200 modelled with a Gaussian process. [Luo Y et al. \(2018\)](#) developed a methodology based on a generative adversarial network for data imputation, in which a modified gate recurrent unit is implemented to model the temporal irregularity of the incomplete time series. [Guo et al. \(2019\)](#) modelled the distribution of multivariate time series by the implementation of a multivariate time series generative adversarial network (MTS-GAN) by proposing the multi-channel convolution into GANs. Then, the missing values were imputed by the formulation of a constrained generation 205 task. Similarly, [Yoon et al. \(2018\)](#) proposed a method named Generative Adversarial Imputation Nets (GAIN), which is an adaption of the generative adversarial networks implemented for imputing missing data. Deep learning methodologies are not regarded in this paper.

The selection of the models to be analysed in the comparative study was applied based on the implementation of a literature review that encompassed the analysis of time series forecasting algorithms and machine learning models. 210 [Hyndman and Athanasopoulos \(2020\)](#) presented a comprehensive analysis of forecasting methods in which time series regression models, exponential smoothing, ARIMA models, and dynamic regression models were implemented. Analogously, [Kotu and Deshpande \(2019\)](#) presented an extensive study of how to implement time series decomposition, smoothing based methods, regression based methods, and machine learning methods. [Kuhn and Johnson \(2016\)](#) intended to provide a guide about predictive modelling processes. Among other aspects, they included 215 a comprehensive description about the data pre-processing step and widely used regression models, such as linear regression and support vector machines.

A summary of the analysed references mentioned within this section is expressed in [Table 1](#), in which a brief description of the methodology implemented together with its utilisation and its limitations are represented. As described, various univariate and multivariate imputation methods have been either analysed or developed to lead to 220 more accurate estimates of incomplete values. In all cases the data imputation method has been implemented in a dataset that contains a certain percentage of missing values. Thus, they did not analyse the capacity of imputing missing values in real time. In addition, most of the studies considered multivariate imputation methods, which implied that predictors were needed to impute missing values. Hence, they did not examine either the possibility of the unavailability of predictors or the probability that the predictors utilised also contained missing values. Various 225 univariate imputation methods were also evaluated, although most of them could only be applied if the time series presented trend and seasonality.

Thus, none of the studies so far has considered the option of imputing incomplete values in real-time to assist instant data-driven decision-making strategies. Furthermore, the applications of those algorithms were very specific, as they did not analyse whether the proposed methodology could work when dealing with different types of datasets. 230 Although several studies have been performed to provide a formal approach for data imputation, very few approaches have been suggested in the maritime industry.

For this reason, a methodology that provides a comparative study of widely used machine learning (ML) and time series forecasting algorithms is performed to assess not only their accuracy to impute incomplete values but also to evaluate their ability to impute these values in real-time.

235 Accordingly, this study provides an extensive analysis of the advantages, disadvantages, and limitations of each implemented technique. The proposed methodology is holistic, and thus includes identification of transient states and data preparation; the latter of which is divided into the data transformation and the correlation analysis steps. Based on the correlation results, both univariate and multivariate imputation methods are implemented if the parameter to be analysed presents at least one predictor, otherwise only the univariate imputation methods are applied. To  
240 evaluate the performance of the suggested models, missing values are generated completely at random and are imputed iteratively by considering time-series cross-validation technique. For each iteration seven metrics are estimated (execution time, MSE, MSLE, RMSE, MAPE, MedAE, and Max Error). Subsequently, once all the values are imputed, the mean of each metric is estimated to assess the overall performance of each model. Thus, the most appropriate imputation method can be determined.

245 The proposed methodology can also be implemented as a framework to determine which algorithm is the most appropriate based on the characteristics of the data and its context, for example when large gaps of missing values need to be treated and the data does not present either trend or seasonality or when either long-term or short-term data are analysed.

**Table 1.** Literature review summary.

| Reference   | Methodology  | Utilisation  | Limitations  |
|---|--|--|--|
| <a href="#">Pratama et al. 2016</a>                 | Review study of conventional and modern imputation procedures.   | -  | -  |
| <a href="#">Chong et al. 2016</a>                   | Comparative study of five imputation methods: <ul style="list-style-type: none"> <li>• Linear regression.</li> <li>• Weighted <math>k</math>-NN.</li> <li>• SVM.</li> <li>• Mean imputation.</li> <li>• Replacing incomplete values with 0.</li> </ul> | Data imputation in time-series sensor data.                                | <ul style="list-style-type: none"> <li>• Linear regression is a multivariate imputation method, and thus its accuracy may decrease if the predictors are not highly correlated with the response. Furthermore, the imputation is only accurate if the mentioned correlation is linear.</li> <li>• Weighted <math>k</math>-NN is another example of a multivariate imputation method, and thus its accuracy hinges on the correlation between the predictors and the response. Moreover, the number of neighbours, <math>k</math>, needs to be estimated, and this may lead to either under-fitting or over-fitting if <math>k</math> is not optimally selected.</li> <li>• SVM also hinges on the correlation between the predictors and the response, its performance may vary based on the kernel selection, and its computational cost is large.</li> <li>• Mean imputation distorts the distribution of the variable and the relationship between variables by reducing estimates of correlation towards zero.</li> <li>• Replacing incomplete values with 0 also distorts the distribution of the variable, which can result in large errors when the incomplete values to impute are far from zero.</li> </ul> |
| <a href="#">Noor et al. 2014</a>                    | Two imputation methods were implemented: <ul style="list-style-type: none"> <li>• Linear interpolation.</li> <li>• Mean imputation.</li> </ul>   | Data imputation in annual hourly monitoring records.                       | <ul style="list-style-type: none"> <li>• Although linear interpolation was the method that presented the most accurate results, it presents some limitations, such as being inaccurate for non-linear functions.</li> <li>• Mean imputation disrupts the inherent structure of the data and degrades the performance of the statistical modelling, as it can lead to large errors in the matrix correlation.</li> </ul>  |
| <a href="#">Balakirshnan and Sangaiah 2018</a>      | Automated framework to impute missing values by applying the context and linear mean (MCL) method  | Data imputation in temperature sensor data.                                | Considering a missing value at time $t$ , this method can only be used if the occurrences at time $t - 1$ and $t + 1$ are available.   |
| <a href="#">Azimi et al. 2019</a>                   | Multiple imputation approach, which utilises short-term data and context and lifestyle data.   | Data imputation in a seven-month monitored data.                           | The proposed method leads to less accurate results when the incomplete data window is small. Moreover, the bias of the estimates may be large if the correlation between context and missing values is not high.   |
| <a href="#">Priya Stella Mary and Arockiam 2017</a> | Methodology based on the assumption that data collected from sensors presents a highly spatial and temporal correlation.   | Data imputation in 5-minute frequency records of air quality sensors data. | The model is only accurate when the relationship between the sensors is strong.  |

|                                     |   |  |   |
|-------------------------------------|---|--|---|
| <a href="#">Fekade et al. 2018</a>  | Probabilistic method to impute missing values from IoT devices by utilising data from analogous sensors. <i>k</i> -Means algorithm is applied to identify neighbours' sensors. Then, Probabilistic Matrix Factorization (PMF) is utilised inside each partition to impute missing values. | Data imputation in data collected from different sensors located in different rooms of a laboratory.   | As PMF is utilised, the complexity increases exponentially with increases in the matrix size. Over-fitting may also occur when the technique is trying to minimise an error that results in a loss of generality. In addition, imputations may not be possible to implement if there are not neighbour sensors available. |
| <a href="#">Bashir and Wei 2018</a> | Method that utilises Vector Autoregressive model (VAR) by combining the Prediction Error Minimisation (PEM) with an EM algorithm. The overall method is named Vector Autoregressive Imputation Method (VAR-IM).   | Data imputation in a dataset including electrocardiogram signals of 290 patients.  | VAR-IM requires the time series to be stationary. Moreover, its performance may not be more accurate than other data imputation methods analysed when the percentage of incomplete values in the dataset is low.  |
| <a href="#">Izonin et al. 2019</a>  | Data imputation method based on the use of the Ito decomposition and the AdaBoost algorithm.  | Data imputation in real data recorded using a certified analyser.  | The proposed approach yields a lesser performance if the data presents anomalies or the size of the sample is not adequately large to train the model.  |
| <a href="#">Liu et al. 2020</a>     | Univariate data imputation method that utilises STL decomposition, named Itr-MS-STLDecImp.  | Data imputation in real-world time series data collected from a Syngas compressor in a real manufacturing plant.   | It is only accurate when dealing with large gaps of data and when the time series presents trend and seasonality.   |
| <a href="#">Bokde et al. 2018</a>   | Method named imputePSF, which is an adjustment of the Pattern Sequence based Forecasting (PSF) algorithm.   | Data imputation in traffic speed time series from a loop detector, in time series of water flow rates generated from hydraulic simulations with EPANET, and nottem dataset that is a twenty year time series of the monthly average air temperature at Nottingham Castle, England. | It is only accurate when the time series presents periodic components, and thus it is not recommended when the time series presents either noisy trends or non-cyclical patterns.   |
| <a href="#">Hegde et al. 2019</a>   | Comparative study of: <ul style="list-style-type: none"> <li>• Probabilistic principal component analysis (PPCA).</li> </ul>  | Data imputation in 116 dental variables.   | Both techniques are multivariate imputation methods, and thus, if the predictors are not highly correlated with the response, the imputation may not be accurate.   |

|                       |   |   |   |
|-----------------------|---|---|---|
|                       | <ul style="list-style-type: none"> <li>Multiple imputation using chained equations (MICE).</li> </ul>   |   |   |
| Hadeed et al. (2020)  | <p>Comparative study of:</p> <ul style="list-style-type: none"> <li>Univariate methods (Mean, Median, Last Observation Carried Forward, Kalman Filter, Random, Markov).</li> <li>Multivariate methods (Predictive Mean Matching, Row Mean Method).</li> </ul> | Data imputation in 20 household with complete 24-hour monitoring data for PM <sub>2.5</sub> .         | <ul style="list-style-type: none"> <li>Univariate imputation techniques may fail to capture expected diurnal or temporal events.</li> <li>Multivariate imputation may present low performances if there are significant differences between households.</li> </ul>  |
| Chivers et al. (2020) | A two-step approach (a binary classification step in tandem with a regression analysis).  | Data imputation in data from a temperate oceanic climate at sub-hourly temporal resolution.           | The comparison between the implemented machine learning models demonstrated their different performances. Ensemble decision tree methods performed well in the classification step, whereas the neural networks performed well in the regression analysis. In no cases $k$ -NN technique was implemented due to the complex and weakly correlated relationship between predictor features and target. Furthermore, the deep learning network of 20 hidden layers was not used either due to over-fitting. |
| Cheliotis et al. 2019 | Hybrid imputation method combining $k$ -NN and MICE algorithms with first-principle knowledge.  | Data imputation in 8 sensors coupled to the turbocharger and to the main engine of a chemical tanker. | As mentioned previously along this table, both $k$ -NN and MICE techniques are multivariate imputation methods, and thus, if the predictors are not highly correlated with the response, the imputation may not be accurate. Also, the number of neighbours, $k$ , needs to be estimated, and this may lead to either under-fitting or over-fitting if $k$ is not optimally selected.   |

### 250 3. Methodology

Having explored the advantages, the disadvantages, and the gaps of various existing imputation techniques developed to lead to more accurate estimates of incomplete values, this section presents a methodology to compare existing data-driven techniques to impute missing values in real-time, which is graphically represented in Fig. 1. The proposed methodology is applied to short-term sensor data, which has been collected from sensors installed on critical marine machine systems and which are stored in a database. Once the data is available, the first step is implemented, the aim of which is the identification of machinery transient states, as datasets may not only include steady operational states but also both manoeuvring and transient states of machinery, which need to be excluded from the analysis. Subsequently, the second step is applied, and thus data is prepared so that it fits adequately into the implemented models. In addition, the correlation analysis in this step is also performed to identify the predictors of each feature being analysed. Once the data pre-processing is completed, the third step, missing values generation, is performed to evaluate the model by applying time-series cross-validation technique, which is performed in step 4. Thus, as more than one missing value is generated in each sample, the time series cross-validation technique is applied to not only evaluate the model accuracy but also to assess their real-time imputation performance, as it is an essential requirement to provide an updated database. Hence, in the steps 5 and 6 the models are implemented for each imputation until all missing values are estimated. The analysed imputation techniques are categorised in the univariate imputation group, which includes mean imputation, time series decomposition techniques, exponential smoothing methods, and ARIMA models, and the multivariate imputation group that encompasses linear regression (Partial Least Squares regression, LASSO regression, Ridge regression, and ElasticNet regression),  $k$ -NN, support vector machines for regression (with linear and RBF kernel), neural networks (with 1, 2, and 3 hidden layers), Vector Autoregressions (VARs), decision tree regressors, and ensemble methods (Bagged trees (with SVR and  $k$ -NN regressors), random forests, and AdaBoost). Univariate methods impute values of a feature by only considering the feature being analysed, whereas multivariate methods impute values of a feature by considering other features that are correlated with the feature being analysed. Thus, multivariate imputation techniques are only implemented if at least one highly correlated predictor is identified for the feature being analysed. The machine learning and time series forecasting models considered in the comparative study are implemented by the utilisation of the Python libraries Scikit-Learn and Statsmodel. To conclude the comparative study, in step 7 a total of seven metrics, recognised as the execution time, the Mean Squared Error (MSE), Mean Squared Logarithmic Error (MSLE), the Root Mean Square Error (RMSE), the Mean Absolute Percentage Error (MAPE), the Median Absolute Error (MedAE), and the Max Error are estimated to evaluate the performance of each implemented model. This step is repeated for every iteration, so that the seven metrics can be

280 estimated for every imputation. Hence, once all the values are imputed, the mean of each metric is estimated to assess their overall performance, and thus determine which imputation method is the most appropriate.

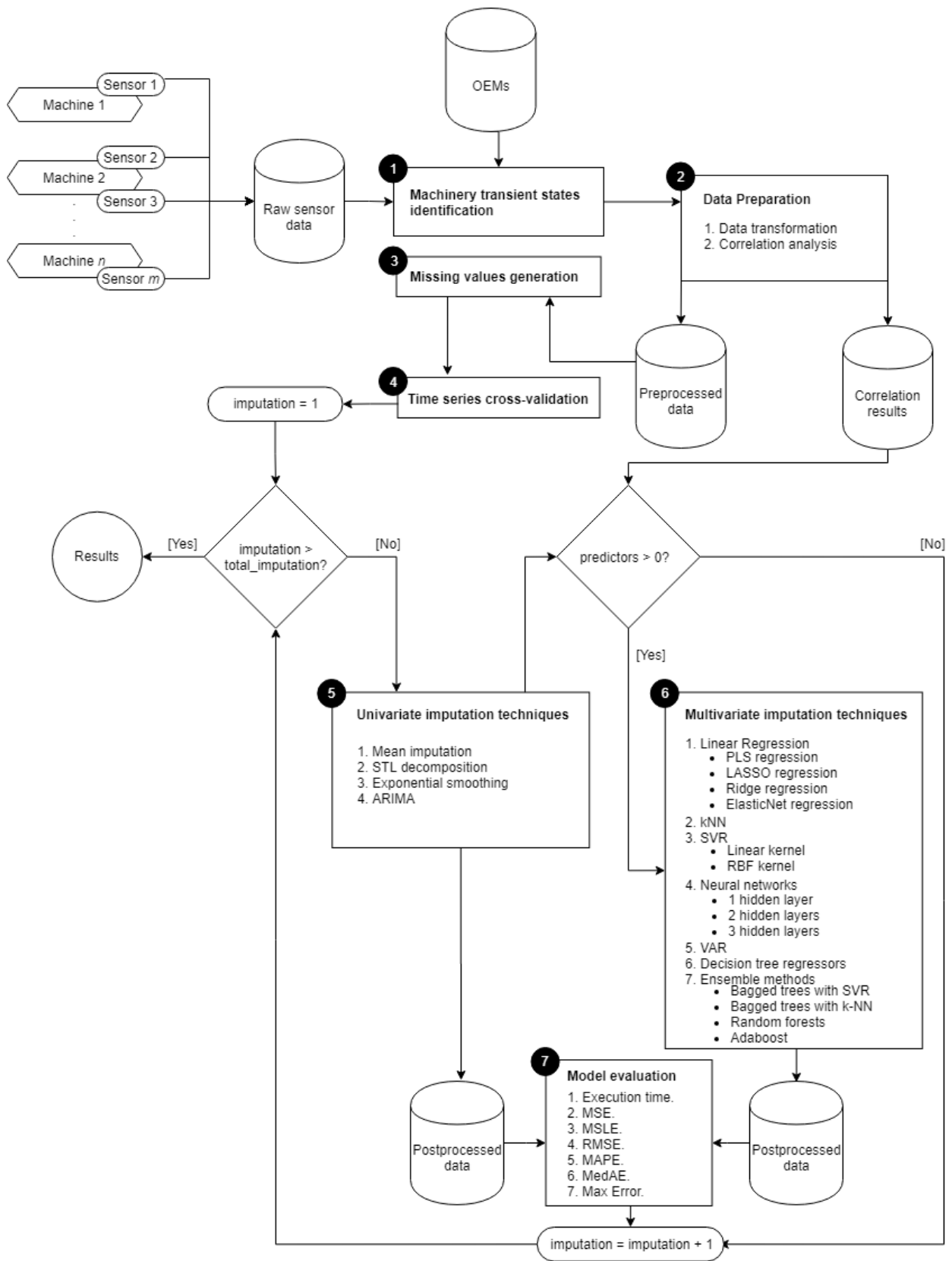


Fig. 1. Graphical representation of the proposed methodology.

285 **3.1. Machinery transient states identification**

Datasets may not only include steady operational states, and thus both manoeuvring and transient states of machinery may also be recorded and included in the datasets. These need to be identified and then discarded. Hence, only those occurrences whose measurements are above a lower threshold limit and below an upper threshold limit are considered. Furthermore, those observations that present a large variability within a period need also to be rejected. Thus, Original Equipment Manufacturers (OEMs) of the systems being analysed need to be consulted in order to define proper steady operational states, and thus define the thresholds adequately.

**3.2. Data preparation**

**3.2.1. Data transformation**

The transformations proposed by Box and Cox are applied to remove distributional skewness (1).

$$x' = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(x) & \text{if } \lambda = 0 \end{cases}, \quad (1)$$

Where  $\lambda$  is a parameter that is determined empirically by training the data and applying Maximum Likelihood Estimation (MLE). Based on the value of  $\lambda$  various widely used transformations can be identified from this family of transformations, such as the square transformation ( $\lambda = 2$ ), the square root transformation ( $\lambda = 0.5$ ), and the inverse transformation ( $\lambda = -1$ ). This transformation is only valid to transform values greater than 0.

Additionally, the data is also standardised, as, for instance, features may not contribute equally in distance-based algorithms if they present different ranges due to the fact that higher ranges will be more influential. Hence, all features are centred by subtracting the mean from all values and scaled by dividing the features by their respective standard deviations (2).

$$z = \frac{x' - \bar{x}'}{s'} \quad (2)$$

As a result of the standardisation, the standardised features present a mean of zero and a standard deviation of one.

**3.2.2. Correlation analysis**

The Pearson's correlation coefficient is determined in order to identify the relationship between two variables. Considering two features,  $x$  and  $y$ , the Pearson's correlation coefficient results from the standardisation of each feature and subsequently estimates the mean after multiplying them (3):

$$\rho = \frac{1}{n-1} \sum_{i=1}^n \frac{x_i - \bar{x}}{s_x} \frac{y_i - \bar{y}}{s_y} \quad (3)$$

If (3) is rewritten by considering the covariance term, which is defined as the mean of the product of the deviations, (4) is obtained.

$$\rho = \frac{\text{Cov}(x, y)}{s_x s_y} \quad (4)$$

Pearson's correlation coefficient always lies between -1 and +1. The strength of the relationship is identified by the magnitude obtained. If the resulting coefficient is either -1 or 1 it indicates that the features are perfectly correlated. Conversely, if Pearson's correlation coefficient is 0 it means that a linear relationship does not exist between them.

315 Although Pearson's correlation coefficient is widely used, it presents some limitations, as it is sensitive to outliers and only captures linear relationships. Thus, the Spearman's rank correlation coefficient, which is based on the rank of the data, is also estimated. Spearman's rank correlation coefficient is not only robust to outliers but is also able to capture certain non-linear relationships.

### 3.3. Missing values generation

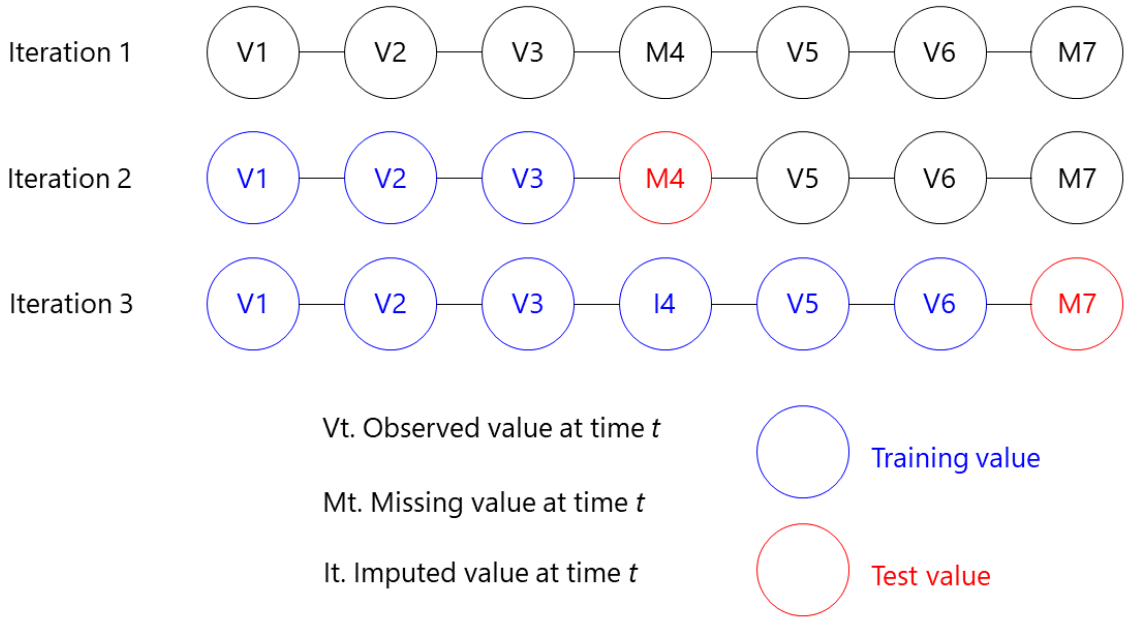
320 To assess adequately the performance of the models to be compared, the source dataset cannot include any missing value. Thus, missing values are generated completely at random to evaluate the accuracy of each model. The percentage of missing values considered for this study is 30%, as it encompasses the range of missing values perceived in the analysed datasets collected from marine machinery systems, which contain from 4.4% to 26% missing values.

### 3.4. Time series cross-validation

325 Some of the models analysed in this study require some parameters that cannot be estimated from the sample and that control the complexity of the model. An inappropriate selection of these hyperparameters may lead to either under-fitting or over-fitting the implemented models, and thus yield inaccurate imputations.

Consequently, time series cross-validation is applied in order to seek to select the optimal hyperparameters that best conduct the learning process. In this version of cross-validation the training set is only formed by those records that were collected prior to the measurements that constitutes the test set. Hence, the training set is only formed by those occurrences recorded prior to the missing value to be imputed and the test set is constituted by this mentioned missing value (Fig. 2). Therefore, the evolution of the feature through time is considered to impute the missing values, and thus the real-time imputation performance of the model can also be evaluated.

330



335

Fig. 2. Time series cross-validation.

### 3.5. Univariate imputation methods

#### 3.5.1. Mean imputation methods

The mean imputation technique is a simple forecasting method widely used to impute missing values by estimating the mean of the sample:

$$\hat{y}_t = \bar{y} = \frac{1}{t-1} \sum_{i=1}^{t-1} y_i, \quad (5)$$

340 where  $\hat{y}_t$  is the predicted value at the current time  $t$ ,  $\bar{y}$  is the mean, and  $y_i$  is the sensor measurement at time  $i$ .

#### 3.5.2. Seasonal and Trend decomposition using Loess method

The Seasonal and Trend decomposition using Loess (STL) method decomposes the time series data into three components that capture trend, seasonal, and residual patterns. The trend component detects the entire evolution of the series along time, whereas the seasonal component captures fluctuation patterns that are repeated through time due to seasonal factors. Those irregularities that do not correspond to either the trend or seasonal components are reflected in the residual component.

Therefore, once these three components are identified the missing values are predicted by considering an additive decomposition and by forecasting the time series components (6).

$$\hat{y}_t = \hat{S}_t + \hat{T}_t + \hat{R}_t, \quad (6)$$

350 where  $\hat{S}_t$  is the forecasted seasonal component,  $\hat{T}_t$  is the forecasted trend component, and  $\hat{R}_t$  is the forecasted residual component. The forecast of the seasonal component is performed based on the assumption that the seasonal component

is unchanging, or slightly changing, and thus its value at time  $t$  can be predicted by implementing a seasonal naïve method. Conversely, the seasonally adjusted component, defined as the addition of forecasted trend and residual components, is estimated by applying a non-seasonal forecasting method, such as the simple exponential smoothing method, which is described in the following section.

### 355 3.5.3. Exponential Smoothing methods

The simple exponential smoothing method is a widely utilised technique when the time series does not distinctly present either trend or seasonality. This results from the fact that the estimation of the predicted values is based on the weighted averages method, in which the most recent observations present the greatest values whilst observations addressed further back decrease exponentially in weight (7).

$$\hat{y}_t = \sum_{i=0}^{t-1} \alpha (1 - \alpha)^i y_{t-i} + (1 - \alpha)^t l_0 \quad (7)$$

360 where  $\alpha$  is the smoothing parameter, the value of which falls between 0 and 1 (inclusive), and  $l_0$  is the least recent observation. Thus, less recent observations present a weight close to 0, and thus do not significantly influence the predicted values, whereas more recent observations have a weight near 1, which indicates that they have a major impact in the imputation of the missing values. Accordingly, the smoothing parameter,  $\alpha$ , and the least recent observation,  $l_0$ , need to be estimated by implementing cross-validation.

365 Holt's linear trend method is an extension of the simple exponential smoothing technique in which the trend of the time series is also considered to impute the missing values. The component form of this method is expressed hereunder.

$$\text{Forecast equation} \quad \hat{y}_t = l_t + b_t \quad (8.1)$$

$$\text{Level equation} \quad l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (8.2)$$

$$\text{Trend equation} \quad b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}, \quad (8.3)$$

where  $l_t$  and  $b_t$  are the estimates of the series and of the trend of the series at time  $t$  respectively, and  $\alpha$  and  $\beta$  are the smoothing parameters for the level and for the trend, which fall between 0 and 1 (inclusive).

370 Analogously, the seasonal component can also be captured by exponential smoothing methods, specifically by the Holt Winder's additive method, in which an additional seasonal equation is considered in the component form.

$$\text{Forecast equation} \quad \hat{y}_t = l_t + b_t + s_{t-m} \quad (9.1)$$

$$\text{Level equation} \quad l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (9.2)$$

$$\text{Trend equation} \quad b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} \quad (9.3)$$

$$\text{Seasonal equation} \quad s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m},$$

### 3.5.4. Autoregressive integrated moving average models

The implementation of these models requires the time series to be stationary. This is presented when both trend and seasonal components are not clearly identified. To determine whether the time series is stationary, both Kwiatkowski-Phillips-Schmidt-Shin (KPSS) and Augmented Dickey-Fuller (ADF) unit root tests are performed. Hence, a time series is considered to be stationary only if both tests reject the hypothesis that the time series is non-stationary.

Accordingly, if the time series is identified as non-stationary differencing is implemented, and thus the difference between consecutive observations is performed until both unit root tests reject the non-stationary hypothesis (10).

$$y'_t = y_t - y_{t-1} \quad (10)$$

The first model analysed in this classification is the autoregressive models, as these models together with moving average models form the ARIMA models. When implementing autoregressive models, the missing value to be imputed is predicted by applying linear combination of prior occurrences (11).

$$y'_t = c + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + \dots + \phi_p y'_{t-p} + \varepsilon_t \quad (11)$$

where  $\varepsilon_t$  is white noise. Analogously, the linear combination of prior predicted errors, known as moving average models, may also be performed to impute missing values. The expression of a moving average model of order  $q$  is described hereunder.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (12)$$

By combining differencing with autoregression and a moving average model, the non-seasonal ARIMA model is constituted (13).

$$y'_t = c + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (13)$$

The orders of the discussed models are determined by minimising the Bayesian information criterion.

### 3.6. Multivariate imputation methods

#### 3.6.1. Linear regression

Linear regression models describe the relationship between a response feature (dependent variable) and one or more explanatory features (independent variables) to predict the values of the response based on the independent variables' values. If only one independent variable is considered, the simple linear regression model is applied. Conversely, the multiple linear regression model is performed if more than one explanatory feature is utilised.

Simple linear regression aims to predict the response variable from the explanatory variable by estimating the regression line that best models the relationship between these two features:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x, \quad (14)$$

where  $\hat{y}$  is the predicted response,  $\hat{\beta}_0$  is the intercept,  $\hat{\beta}_1$  is the slope or regression coefficient, and  $x$  is the explanatory variable.

400 The best adjusted regression line is obtained by the implementation of the Ordinary Least Squares (OLS) regression method, which estimates the regression line that minimises the sum of squared residual values, being a residual value the difference between the observed and the predicted value ( $e_i = y_i - \hat{y}_i$ ).

A generalisation of the simple linear regression model is the multiple linear regression model, where the response variable is related to  $k$  explanatory variables ( $x_1, x_2, \dots, x_k$ ):

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + e. \quad (15)$$

405 The predictors considered in the multiple linear regression may be strongly correlated, which implies both high variability and instability of the solution provided by the OLS. In such cases, the partial least squares (PLS) is suggested. PLS seeks linear combinations between predictors, also named components, which are selected to not only summarise the variation of the predictors at maximum but also ensure that the estimated components present maximum correlation with the response. Hence, PLS is applied prior to linear regression model creation when the independent  
410 variables are highly correlated to obtain the components that will be utilised as predictors. To estimate the optimal number of components to be retained cross-validation is applied.

Collinearity between predictors can also be treated with biased models, as the entire MSE can be reduced when a trade-off between the bias and the variance is applied due to the fact that the variance can be reduced by increasing the bias slightly. These biased regression models can be created by adding a penalty, which regularises the  
415 parameter estimates, to the sum of the squared error. Ridge regression considers the addition of a second-order penalty on the parameter estimates (16).

$$SSE_{L_2} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2, \quad (16)$$

where  $\lambda$  is the penalty parameter. The greater this value is the more the method shrinks the estimates towards 0. An alternative to ridge regression is the Least Absolute Shrinkage and Selection Operator (LASSO) model, in which the absolute value of each parameter is added to regularise the parameter estimates (17).

$$SSE_{L_1} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\beta_j|. \quad (17)$$

420 By applying this type of regularisation, some parameters may be set to 0, and thus these are not considered in the penalised regression model. Hence, LASSO regression method is not utilised to improve the model accuracy only, but also to implement feature selection.

Additionally, an extended version of Ridge and LASSO regression methods that combine the two penalties is the ElasticNet regression method, the SSE of which is expressed hereunder.

$$\text{SSE}_{\text{Enet}} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^p \beta_j^2 + \lambda_2 \sum_{j=1}^p |\beta_j|. \quad (18)$$

425 The main advantage of this model is that the regularisation is enabled effectively by adding ridge penalty and the feature selection is applied by applying the LASSO penalty in a quality manner. As the entire penalised regression models considered in this study present hyperparameters, cross-validation is applied to achieve their optimal values.

### 3.6.2. *K*-Nearest Neighbors

430 A non-linear regression method widely used to impute missing values is the *k*-Nearest Neighbors (*k*-NNs), which aims to predict the missing values by considering *k*-closest records from the training set. Thus, the value of the response is obtained by estimating the mean of the *k*-nearest neighbors, which are selected by calculating the Euclidean distance between the samples (19).

$$d_{\text{Euclidean}} = \left( \sum_{i=1}^p (x_{1i} - x_{2i})^2 \right)^{\frac{1}{2}}, \quad (19)$$

where  $x_1$  and  $x_2$  are the analysed instances. The utilisation of the distance metric in this method can disrupt the prediction of the response if any predictor value is missing, as then the distance cannot be estimated. Hence, to address 435 this problem, either the instances that contain missing values are excluded from the analysis or a univariate imputation method, such as mean imputation, is implemented prior to the *k*-NNs model to impute the missing values. However, to assess the accuracy of the model, it is considered the predictors in the analysed sample do not contain missing values. Additionally, another aspect to be addressed in the *k*-NNs method is the number of neighbours to be considered, which, to avoid over-fitting, is estimated by the square root of the total number of occurrences of the 440 sample.

An enhancement of this method is the weighted *k*-NNs, in which a weight is added to the selected neighbours to regularise their contribution into the response prediction. Thus, those neighbours that are closer present more weight, and thus contribute more to the response prediction, whereas those that present a greater distance contribute less.

### 445 3.6.3. Support vector machines

Support Vector Machines (SVMs) is another widely used highly flexible non-linear modelling method, which was initially implemented as a classification model. In SVMs for regression, the model is formed by only those data points the residuals of which present an absolute difference greater than a given threshold, denoted as  $\epsilon$ . The coefficients of the SVM regression model minimise

$$C \sum_{i=1}^n L_{\epsilon}(y_i - \hat{y}_i) + \sum_{j=1}^p \beta_j^2, \quad (20)$$

450 where  $L_{\epsilon}(\cdot)$  is the  $\epsilon$ -insensitive function, and  $C$  is the cost parameter, which penalises large residuals. The prediction function of the SVM can be written as

$$f(\mathbf{u}) = \beta_0 + \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{u}), \quad (21)$$

where  $K(\cdot)$  is the kernel function. Among the kernel functions that encompass non-linear functions of the predictors, the Radial Basis Function (RBF) is considered in this study (22).

$$K(\mathbf{x}_i, \mathbf{u}) = \exp(-\sigma \|\mathbf{x}_i - \mathbf{u}\|^2) \quad (22)$$

In addition, the linear kernel is also considered (23).

$$K(\mathbf{x}_i, \mathbf{u}) = \mathbf{x}_i' \mathbf{u} \quad (23)$$

455 This technique also presents hyperparameters, which are the cost parameter  $C$  and the threshold  $\epsilon$ . Additionally, the RBF also presents the  $\sigma$  parameter, the optimal value of which also needs to be estimated. Thus, cross-validation is also applied in this case to achieve their optimal values.

### 3.6.4. Neural Networks

460 Neural Networks (NNs) are another example of non-linear regression methods, inspired by biological neural networks, in which the response is modelled by hidden units. These hidden units are defined as an intermediary set of unobserved variables constituted by either linear or non-linear combinations of the predictors, the outcomes of which are combined again to either be utilised as inputs for the subsequent hidden layer or to predict the response.

To estimate the optimal parameters that constitute the combinations applied along the NN, the weight decay penalisation method is utilised to regularise the model by adding a penalty for large regression coefficients. Hence, 465 by applying this regularisation over-fitting is moderated.

### 3.6.5. Vector autoregressive models

470 As ARIMA models, the VAR models are limited to be accurate only by those features that present to be stationary. This is due to the fact that the VAR models are a generalisation of the univariate autoregressive models, in which not only a linear combination of prior occurrences of the analysed feature is considered but also bi-directional relationships between features are also included, as the overall features are considered as endogenous. Hence, the VAR models are able to predict a vector of time series iteratively by generating predictions for each feature included in the model, as it is expressed in (24) for a VAR model of order 1 and dimension 2.

$$\hat{y}_{1,t} = \hat{c}_1 + \hat{\phi}_{11,1} \hat{y}_{1,t-1} + \hat{\phi}_{12,1} \hat{y}_{2,t-1} \quad (24.1)$$

$$\hat{y}_{2,t} = \hat{c}_2 + \hat{\phi}_{21,1} \hat{y}_{1,t-1} + \hat{\phi}_{22,1} \hat{y}_{2,t-1} \quad (24.2)$$

The order of the model is determined by minimising the Bayesian information criterion.

### 3.6.6. Decision trees regressors

475 Decision trees regressors are another example of predictive models in which the basis of the prediction is established by partitioning the feature space into subspaces in an iterative manner. The tree begins with the root node, constituted with the first partition that splits the data into disjoint sets, which in turn are divided into smaller partitions. Hence, subsequent children nodes are split until the optimal number of partitions is achieved.

480 Among all the techniques utilised for constructing regression trees, the classification and regression tree methodology is one of the most widely utilised. For regression, from the overall sample, this methodology begins by partitioning the data into two groups, of which the sums of the squared errors are minimised (25).

$$SSE = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2, \quad (25)$$

where  $\bar{y}_1$  and  $\bar{y}_2$  are the mean of the groups of data  $S_1$  and  $S_2$  respectively. Iteratively, the subsequent groups of data are split until the optimal number of partitions are achieved. To avoid over-fitting, a complexity parameter can be included to penalise the error rate by utilising the tree size (26).

$$SSE_{c_p} = SSE + c_p n_{\text{terminal nodes}}, \quad (26)$$

485 where  $c_p$  is the complexity parameter.

### 3.6.7. Ensemble methods

The utilisation of single regression trees is likely to present sub-optimal predictive performance due to their limitations, such as their instability. Therefore, ensemble methods are suggested, as they tend to present better performance.

490 An example of which is the bootstrap aggregation trees, also referred as bagged trees, in which bootstrapping is implemented in tandem with a regression model. Hence, this ensemble method generates  $m$  samples, obtained by implementing bootstrapping from the original data. Then, an unpruned tree model is trained on each resulting sample, the predictions of which are averaged to obtain the resulting bagged model's prediction.

495 Random forest is another example of ensemble method that adds randomness into the learning process to reduce correlation among predictors. Analogous to bagged trees, the method generates  $m$  samples, obtained by implementing bootstrapping from the original data. However, the tree model on each sample is trained by applying random split selection, in which the tree is modelled by utilising a random subset of the top  $k$  predictors at each split in the tree. The resulting random forest model's prediction is obtained by estimating the average of the samples' predictions.

500 Additionally, boosting methods are another class of ensemble methods, the origin of which initially was to solve classification problems. The basis of these types of methods is the recursive modelling of compositions, in which

each subsequent model learns by utilising the error information identified in the previous one. The adaptive boosting technique, also referred as AdaBoost, is an example of a boosting method, which implements weight adjustment procedures based on the errors of the current predictions. Hence, in each iteration, larger weights are assigned to more complicated predictions so that the succeeding tree can target them in more detail.

### 3.7. Model evaluation

Seven metrics are utilised to assess the performance of the imputation techniques implemented. These metrics are estimated every iteration. Hence, once all the values are imputed, the mean of each metric is estimated to assess their overall performance, with the exception of the Median Absolute Error (MedAE) and the Max Error regression metrics in which the median and the maximum value selection have been implemented instead.

#### 3.7.1. Execution time

The first metric estimated is the execution time, which is obtained by applying the difference between the function end time and the function start time. Thus, we can obtain the time utilised to impute the missing value at each iteration for each model.

#### 3.7.2. Mean Squared Error (MSE)

MSE is obtained by estimating the mean of the sum of the squared errors, as defined in (27).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (27)$$

where  $n$  corresponds to the number of samples, and  $y_i$  and  $\hat{y}_i$  refers to the  $i$ -th occurrence of the observed and the predicted values, respectively. MSE is probably the most generally utilised loss function for regression, as if  $\frac{1}{n}$  is discarded from the equation, the Least Square Errors function ( $L^2$ ) is obtained. As the errors are squared, this metric penalises larger errors, which makes MSE sensitive to outliers.

#### 3.7.3. Mean Squared Logarithmic Error (MSLE)

MSLE refers to the expected value of the squared logarithmic error (28).

$$\text{MSLE} = \frac{1}{n} \sum_{i=1}^n (\ln(1+y_i) - \ln(1+\hat{y}_i))^2 \quad (28)$$

where  $\ln(x)$  refers to the natural logarithm of  $x$ . MSLE penalises under-predicted estimates more than over-predicted ones, and thus asymmetry is introduced in the error curve.

#### 3.7.4. Root Mean Square Error (RMSE)

RMSE is a type of scale-dependent error, which indicates that the estimated errors are on the same scale as the observations, and its value is obtained by estimating the squared root of MSE (29).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (29)$$

### 3.7.5. Mean Absolute Percentage Error (MAPE)

530 Another type of error is the percentage error, which is given by (30).

$$p_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (30)$$

The mean absolute percentage error (MAPE), defined as the mean of the sum of the percentage errors (31), is a widely used percentage error metric, which is also computed for model evaluation, as it is unit-free. Conversely, one of its drawbacks is that its value is undefined if the observed value is 0.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n p_i \quad (31)$$

### 3.7.6. Median Absolute Error (MedAE)

535 MedAE is computed by estimating the median of all absolute differences between the observed and the predicted occurrences (32).

$$\text{MedAE} = \text{median}(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n|) \quad (32)$$

Contrary to the other metrics introduced in this section, MedAE is robust to outliers due to the consideration of the median performance, which makes this regression metric particularly interesting.

### 3.7.7. Max Error

540 The maximum residual error is also computed to capture the worst-case error.

$$\text{Max Error} = \max(|y_i - \hat{y}_i|) \quad (33)$$

## 4. Results

In this section, a DMD-MAN B&W 6S50MC-C main propulsion engine of a cargo vessel is employed to demonstrate the methodology presented above. This is a camshaft controlled two-stroke engine, which utilises super long stroke to bore ratio, constituted by a total of 6 cylinders with 50 centimetres diameter pistons. The rotational speed, the power and the fuel flow rate of this main engine are analysed in conjunction with 4 more parameters obtained from sensors that are installed on the lubrication oil system, on the jacket cooling water system, on the turbocharger, and on the scavenge air receiver with a 1 minute frequency (Table 2).

**Table 2.** Main engine system monitored parameters.

|             | Parameter        | Units |
|-------------|------------------|-------|
| Main Engine | Rotational Speed | r/min |
|             | Power            | kW    |

|                             |  |       |
|-----------------------------|--|-------|
|                             | Fuel Flow Rate                         | tn/hr |
| Lubrication Oil System      | Inlet Pressure                         | bars  |
| Jacket Cooling Water System | Inlet Pressure                         | bars  |
| Turbocharger                | Turbine Lubricating Oil Inlet Pressure | bars  |
| Scavenge Air Receiver       | Scavenging Air Pressure                | Bars  |

A sample that contains a total of 2,000 records of each feature is obtained from the source samples. All the records refer to the steady operational states of machinery. Thus, manoeuvring, and transient states of machinery are not included in this analysis.

Fig. 3 visualises the evolution of the main engine rotational speed throughout a 34-hour sample. Overall, the rotational speed presents a low variability over time. There is a large steady state that initiates at the first instant and persists over half the recorded time where the values are stabilised around 105.0 r/min. Subsequently, there is a slight adjustment where the rotational speed decreases to approximately 102.5 r/min. This state remains for roughly 300 minutes, and suddenly an abrupt increase in the revolutions is perceived. This is when the maximum value of the time series is achieved, which presents a value greater than 110.0 r/min. The state remains for several minutes and then the rotational speed is decreased in three slight phases until the minimum state is recorded; its values being lesser than 100.0 r/min. This state begins at 1,500 minutes and remains constant until the end of the time series. The abrupt changes refer to small adjustments that are applied due to the contractual agreements between the charterer and the shipowner in relation to the vessel speed and the fuel oil consumption per day. Hence, these adjustments can also be identified in other analysed parameters that are related to the vessel speed and the fuel oil consumption, such as the main engine power, the main engine fuel oil consumption, and the scavenging air pressure of the scavenge air receiver.

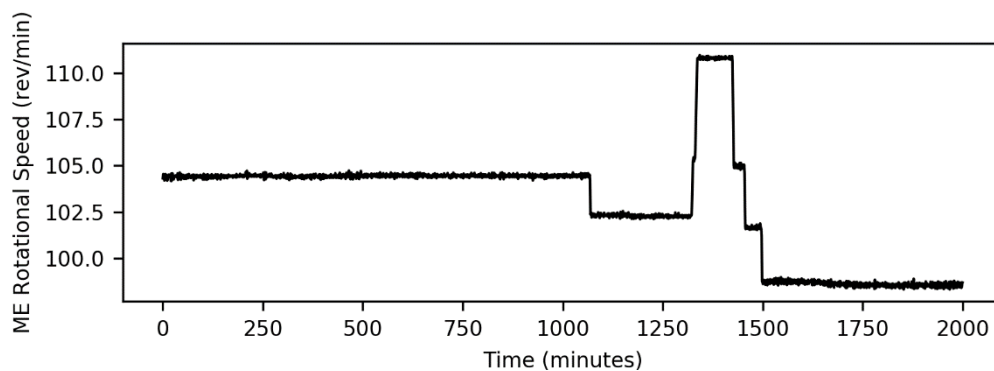


Fig. 3. Time series plot of the main engine rotational speed.

Analogously, the main engine power presents a similar evolution. However, a greater variability can be observed, as, for example, in the first steady state of the time series a peak is produced around minute 250 (Fig. 4). However, the variability along the time series is still considered low.

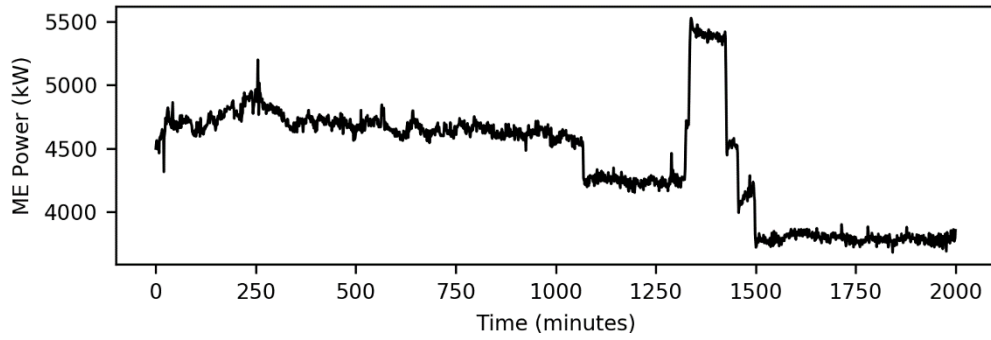
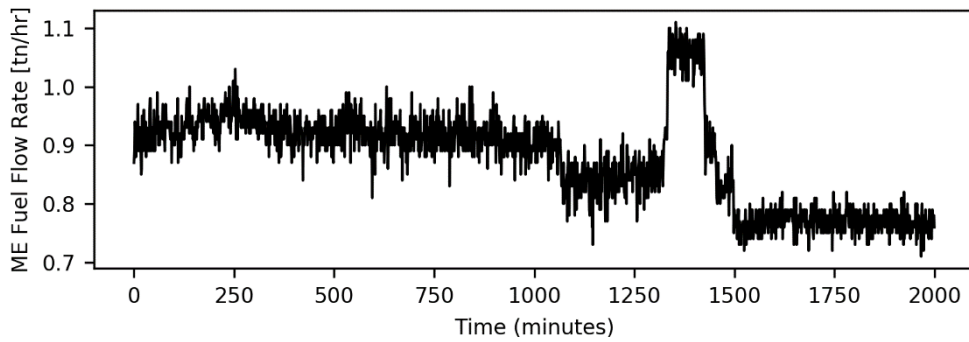


Fig. 4. Time series plot of the main engine power.

570 Furthermore, time series plots of both the main engine fuel flow rate and the scavenging air pressure of the scavenge air receiver system are shown in Fig. 5 and Fig. 6. These indicate similar progressions, as both features are critical for the internal combustion, which produces the linear movement of the piston that promotes the rotating movement of the crankshaft in order to generate power (Wärtsilä, 2020).



575 Fig. 5. Time series plot of the main engine fuel flow rate.

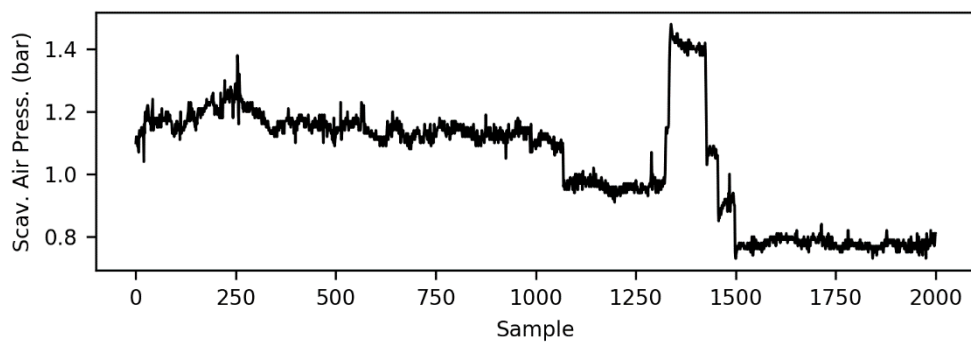


Fig. 6. Time series plot of the scavenging air pressure of the scavenge air receiver system.

Two main steady states can be identified in the lubrication oil inlet pressure time series. The first steady state initiates at the first instant and persists for more than 1,250 minutes where the values are stabilised between 2.39 and 2.40 bars. Subsequently, there is a slight adjustment where the inlet pressure decreases to approximately 2.35 bars (Fig. 7).

580

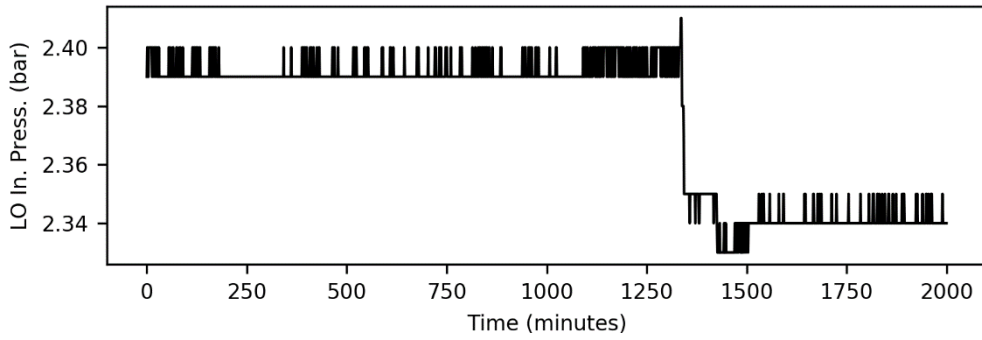


Fig. 7. Time series plot of lubrication oil inlet pressure.

Conversely, the inlet pressure of both the jacket water cooling and the turbocharger lubrication oil systems present a steady state where neither a slight nor an abrupt change is produced. Thus, in the case of the jacket water cooling system inlet pressure the values remain between 3.50 and 3.65 bars (Fig. 8), while the values of the turbocharger lubrication oil inlet pressure fluctuate between 1.8 and 2.6 bars (Fig. 9).

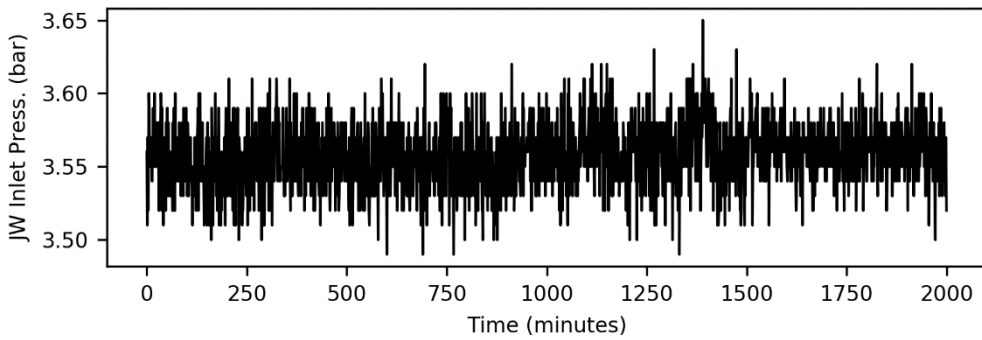


Fig. 8. Time series plot of the jacket water cooling system inlet pressure.

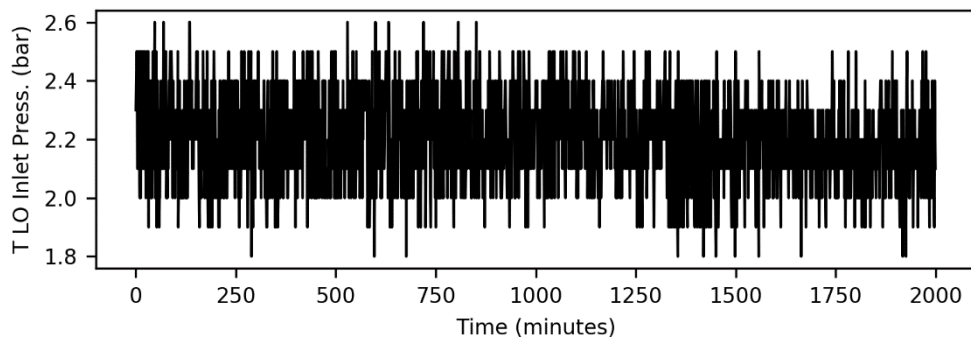


Fig. 9. Time series plot of the turbine lubrication oil inlet pressure of the turbocharger.

**Table 3.** Descriptive statistics of the monitored features.

|      | Main Engine      |            |                        | Lubrication Oil    | Jacket Cooling Water | Turbocharger            | Scav. Air Receiver.    |
|------|------------------|------------|------------------------|--------------------|----------------------|-------------------------|------------------------|
|      | Speed (rev./min) | Power (kW) | Fuel Flow Rate (tn/hr) | Inlet Press. (bar) | Inlet Press. (bar)   | T LO Inlet Press. (bar) | Scav. Air Press. (bar) |
| Mean | 102.95           | 4421.09    | 0.88                   | 2.38               | 3.56                 | 2.20                    | 1.04                   |

|      |        |         |      |      |      |      |      |
|------|--------|---------|------|------|------|------|------|
| Std. | 2.98   | 438.71  | 0.08 | 0.02 | 0.02 | 0.16 | 0.18 |
| Min. | 98.32  | 3676.14 | 0.71 | 2.33 | 3.49 | 1.80 | 0.73 |
| 25%  | 98.94  | 3878.76 | 0.80 | 2.34 | 3.54 | 2.10 | 0.83 |
| 50%  | 104.37 | 4601.76 | 0.90 | 2.39 | 3.56 | 2.20 | 1.12 |
| 75%  | 104.47 | 4697.83 | 0.93 | 2.39 | 3.57 | 2.30 | 1.16 |
| Max. | 110.96 | 5528.74 | 1.11 | 2.41 | 3.65 | 2.60 | 1.48 |

Table 3 illustrates the descriptive statistics results of the seven monitored parameters being analysed. A total of three parameters from the main engine system are considered. The main engine rotational speed has a mean of 102.95 r/min and a standard deviation of 2.98 r/min, which indicates that the feature presents low variability. Its median is 104.34 r/min, which differs from the mean value, the median being greater than the mean, and thus indicating that the data distribution is not only asymmetric but is also skewed to the left. The main engine power has a mean of 4421.09 kW and a standard deviation of 438.71 kW. In this case, the feature presents more variability, although it is still considered as low, and the median, which is 4601.76 kW, is greater than the mean, indicating that feature presents a left-skewed distribution. The third and last component being analysed from the main engine system is the fuel flow rate. Its mean is 0.88 tn/hr and its standard deviation is 0.08 tn/hr, thus implying that this parameter also presents low variability. It has a median of 0.90 tn/hr, which slightly differs from the mean, and thus the data distribution is considered as asymmetric, although its asymmetry is not as significant as the features previously analysed.

Only one parameter is considered with respect to the lubrication oil system. This is the inlet pressure, which presents a mean of 2.38 bars and a standard deviation of 0.08 bars, indicating low variability along the time series. Furthermore, the median is also estimated, which has a value of 2.39 bars, and thus presents the same value as the third quartile, which implies that the distribution is skewed to the left.

From the jacket cooling water system, the inlet pressure parameter is analysed. It has a mean of 3.56 bars, which is equal to the median, and thus indicating that the distribution is symmetric. The standard deviation is 0.02 bars, which indicates its low variability. Analogously, the turbine lubricating oil inlet pressure of the turbocharger has a mean of 2.20 bars, which also coincides with its median, indicating that the distribution is also symmetric. Its standard deviation is 0.16 bars, which also implies that its variability is low.

One parameter of the scavenge air receiver is also analysed. This is the scavenging air pressure that has a mean of 1.04 bars and a standard deviation of 0.18 bars. Again, the feature presents low variability. With the respect to the symmetry, the median differs from the mean, the median being greater than the mean, and thus indicating that the distribution is skewed to the left.

All the features excluding the inlet pressure of the jacket cooling water system and the turbine lubrication oil inlet pressure of the turbocharger, which present nearly a symmetric distribution, are slightly or highly skewed. Hence,

620 the Box-Cox transformation is applied to all features in order to remove their skewness, and also standardisation is implemented to avoid unequal contribution of the features in the analysed models.

To analyse the relationship between features, the Pearson's and Spearman's rank correlation coefficients are estimated for each possible correlation. The absolute values of the obtained results are contained in two different matrices (Pearson's correlation coefficient matrix and Spearman's rank correlation coefficient matrix) and represented  
 625 by displaying heatmap plots (Fig. 10 and Fig. 11).

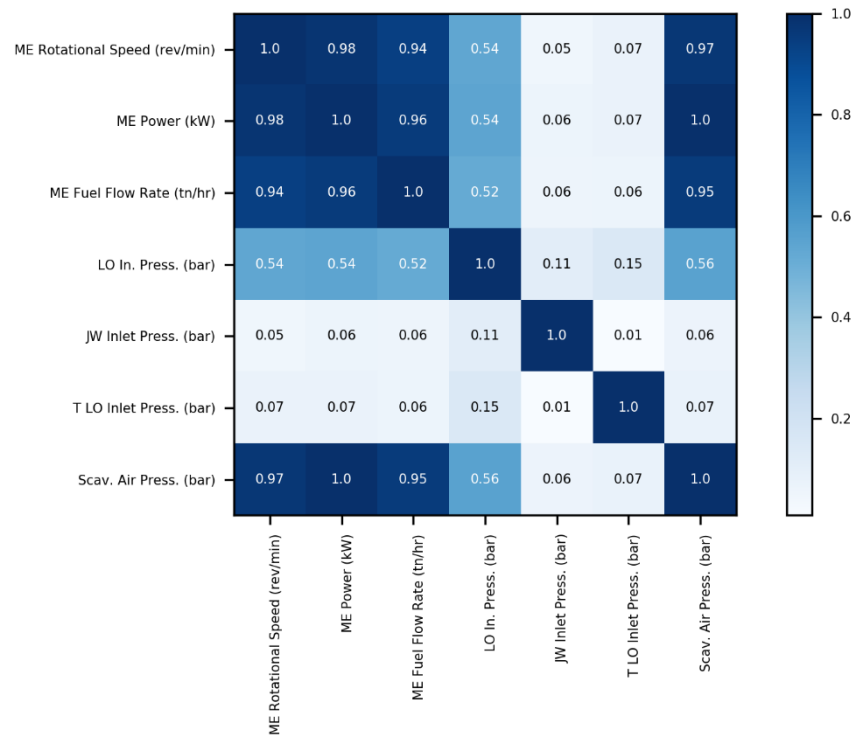


Fig. 10. Heatmap plot of Pearson's correlation coefficient matrix.

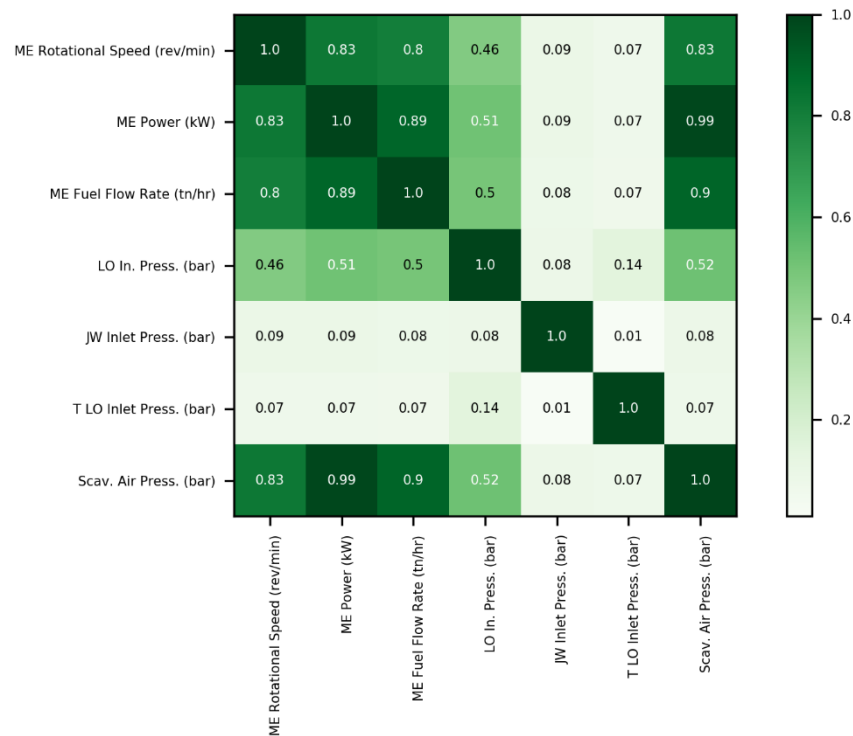


Fig. 11. Heatmap plot of Spearman's rank correlation coefficient matrix.

630 The rotational speed of the marine engine is highly correlated with the engine power. Additionally, it is also correlated with the fuel flow rate and the scavenge air pressure, as these parameters influence the engine combustion. Thus, the power of the marine engine presents a strong relationship with the engine rotational speed, the fuel flow rate, and the scavenge air pressure. The fuel flow rate and the scavenge air pressure are not only correlated with the rotational speed and the engine power but also between themselves, as they present a relationship derived from the conditions needed along the combustion process. The inlet pressure of the jacket cooling water system is not correlated with any presented feature, as it does not have any contact with any other analysed system. Analogously, the turbine lubrication oil inlet pressure of the turbocharger neither influence nor are influenced by any analysed feature. Regarding the inlet pressure of the lubrication oil system, some relationships can be observed with the main engine power, for example, although they are not considered significantly strong, and thus are not considered for this study.

640 Hence, after the cross-reference between data-driven correlation analysis and engineering knowledge the resulting correlation matrix is presented in Table 4, where only those features that have at least one relationship with another feature are presented.

**Table 4.** Correlation matrix of the monitored features.

|                            | Main Engine                |            |                        | Scav. Air Receiver     |
|----------------------------|----------------------------|------------|------------------------|------------------------|
|                            | Rotational Speed (rev/min) | Power (kW) | Fuel Flow Rate (tn/hr) | Scav. Air Press. (bar) |
| Rotational Speed (rev/min) |                            | •          | •                      | •                      |
| Power (kW)                 | •                          |            | •                      | •                      |
| Fuel Flow Rate (tn/hr)     | •                          | •          |                        | •                      |
| Scav. Air Press. (bar)     | •                          | •          | •                      |                        |

645 The imputation results of the main engine power can be observed in Fig. 12, in which a scatterplot between the observed and the imputed values is performed for each analysed model. In this case, it is worth mentioning that although all seven parameters have been considered for all above mentioned models, the main engine power is presented and analysed only due to the present paper extent limitations.

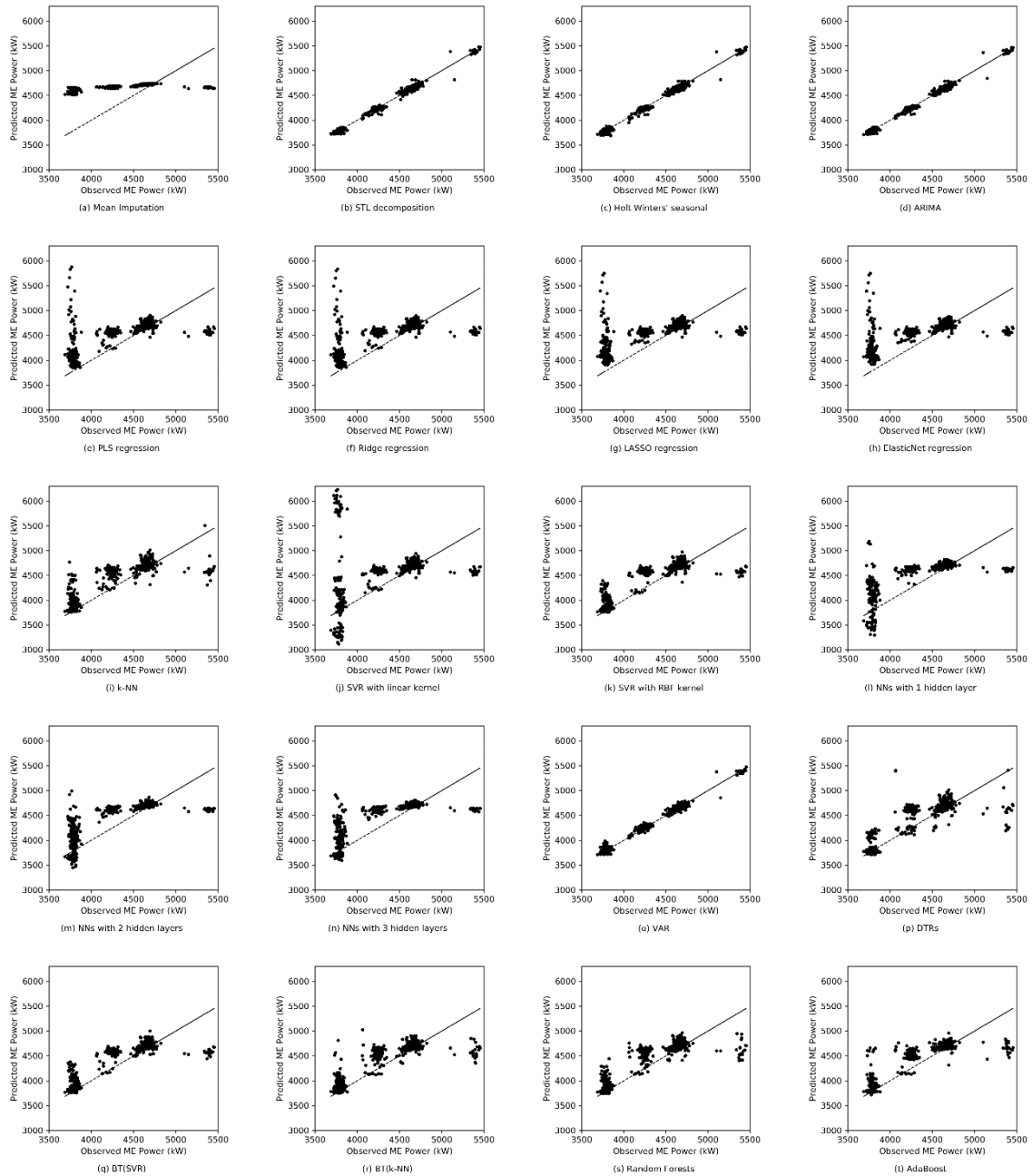


Fig. 12. Comparison between observed values of the main engine power parameter and imputed values by the implementation of the machine learning and time series forecasting models.

650 The first univariate imputation method analysed is the mean imputation, considered a naïve method that imputes incomplete values with the mean of the current analysed sample. Mean imputation yielded one of the worst results of the entire modelled techniques, leading to the most biased estimates when imputing missing values identified in the main engine fuel flow rate (with a MAPE of 9.17% and MedAE of 0.066 tn/hr, see Table 7), and the scavenging

air pressure of the scavenge air receiver (with a MAPE of 19.97% and MedAE of 0.18 bars, see [Table 11](#)). This lack  
655 of accuracy is due to the distortion of the parameter distribution, as expressed in [Fig. 12 \(a\)](#), where the incomplete  
values are imputed with nearly the same value, which corresponds with the mean of the current sample being analysed,  
and thus obtaining bias estimates. Hence, by disrupting the distribution of the variable the relationship between  
variables is also affected, and thus reducing correlation estimates towards zero. Additionally, if the nature of the  
incomplete values is determined by either missing at random (MAR) or not missing at random (NMAR) and the mean  
660 imputation is utilised, the mean estimate may be biased. For instance, if the sensor is not recording when the machine  
status is not operational and this period is imputed with a sample that contains operational condition data, the imputed  
sample does not correspond with the real machinery condition, and thus leads to inaccurate energy operations of  
marine systems due to the utilisation of a biased dataset. However, this univariate imputation method is extremely  
easy to interpret and implement with an execution time of the order of milliseconds.

665 The second univariate imputation method analysed is the Seasonal and Trend decomposition using Loess  
(STL) technique, based on the forecasting of the seasonal component and the seasonally adjusted component. This  
imputation method yielded one of the most accurate results, which is the most accurate imputation technique in the  
inlet pressure parameter of the lubrication oil system (with an execution time of 0.02 seconds and a RMSE of 0.002  
bars, see [Table 8](#)). STL decomposition is more appropriate to implement than other decomposition techniques, such  
670 as classical decomposition and X11 decomposition. It is robust to outliers, the trend smoothness can be regulated, any  
type of seasonality can be considered, and it can be adjusted over time. Also, as expressed in [Fig 12 \(b\)](#), nearly all  
predicted estimates are similar to the observed occurrences, and its execution time is low. However, it can only be  
implemented if the time series presents trend and seasonality and the seasonal period needs to be indicated, which  
may derive bias estimates if it is not optimally selected.

675 Another univariate imputation technique implemented is the Holt-Winters' seasonal method, which is a type  
of exponential smoothing method. This method leads to the one of the most accurate results in the main engine  
rotational speed parameter (with a MAPE of 0.13% and a MedAE of 0.08 r/min, see [Table 5](#)) and, as observed in [Fig.](#)  
[12 \(c\)](#), nearly all predicted estimates are similar to the observed occurrences, thus obtaining similar results as STL  
decomposition technique. Although only Holt-Winters' seasonal method is considered in this study, any exponential  
680 smoothing method can be implemented, which are easy to interpret and apply. Furthermore, one crucial advantage of  
this method is that it indicates recent observations as more significant than earlier observations. Additionally, different  
exponential smoothing methods can be applied based on the characteristics of the data. Simple exponential smoothing  
is considered, therefore, if short-time data is utilised and it does not present trend and seasonality. Conversely, Holt's  
linear trend method is more appropriate if a trend is identified in the data, whilst Holt-Winters' seasonal method is  
685 suggested when the time series presents both trend and seasonality. However, this advantage implies a drawback,

which is the proper selection of the exponential smoothing method. Tuning parameters, such as the smoothing parameter, also need to be optimally selected.

The last univariate imputation method analysed encompasses the autoregressive integrated moving average (ARIMA) models, which lead to the best results in all the analysed parameters (see [Table 5](#) – [Table 11](#)). For example, a MedAE of 0.076 r/min and a Max Error of 2.4 r/min were obtained when imputing the missing values of the main engine rotational speed parameter. Hence, one of its main advantages is the high accuracy when imputing incomplete values in short-term data, as expressed in [Fig. 12 \(d\)](#). In addition, it is applicable to nearly all types of time series. However, it only captures linear relationships. It needs more data than other univariate imputation methods and the orders of the differencing, the autoregression and the moving average model need to be optimally identified. In relation to its execution time, the algorithm presents more computational cost than any other univariate imputation method analysed, and thus its execution time is relatively high, with respect to the other execution times analysed.

Subsequently, the multivariate imputation methods are analysed. The first group implemented comprises the linear regression methods, which consist of partial least squares and penalised methods (ridge regression, LASSO regression, and ElasticNet regression). As these methods impute incomplete values of a feature by considering other features that are correlated with the feature being analysed, only those parameters that present predictors are analysed, which are the main engine rotational speed, main engine power, main engine fuel flow rate, and the scavenging air pressure of the scavenge air receiver. In all cases, the results obtained after implementing each model are similar, being nearly equal in the case of the main engine fuel flow rate (with a MSE of  $0.005 \text{ (tn/hr)}^2$ , see [Table 7](#)). In the case of the main engine rotational speed, the Ridge regression is the linear regression technique that yields better results (with a RMSE of 1.92 r/min and a MedAE of 1.72 r/min, see [Table 5](#)), whereas in the case of the main engine power the PLS is the linear regression method with the best performance (with a RMSE of 272.63 kW and a MedAE of 182.38 r/min, see [Table 6](#)). As observed in [Fig. 12 \(e\)](#) – [Fig.12 \(h\)](#), the predicted values are more dispersed in comparison to the ones estimated by the univariate imputation techniques that lead to the best results. In particular, this lack of accuracy can be easily observed in the outermost values, those farthest from the line, whereby the lesser values to be predicted in the fourth steady state (see [Fig. 4](#)) present the most extreme values; greater than 6,000 kW. Whereas the observed values present a value lower than 4,000 kW. This can be due to the fact that the current time series analysed does not contain enough data to identify the current steady state, and thus the predicted values are extrapolated, leading to a poor performance. Furthermore, linear regression methods are sensitive to outliers and only capture linear relationships between the response and the predictors. However, it is considered one of the best methods due to interpretability and complexity when dealing with a linear relationship, and its execution time is low.

Although all the relationships analysed are linear, non-linear models are also analysed. The first non-linear technique implemented is the  $k$ -Nearest Neighbors ( $k$ -NN), which aims to predict the incomplete values by considering

$k$ -closest records from the training set. Hence, the main drawbacks of this method are the selection of the optimal number of neighbours and the need to employ the entire training set every time an incomplete value is identified. While the execution time in this inquiry is certainly low when implementing  $k$ -NN, its performance may be degraded when the analysed sample is considerably large, when the number of dimensions considered are high, and/or when either noisy data or outliers are presented in the analysed sample. However, in all studied cases, the  $k$ -NN method yields better results than linear regression models, as expressed in Tables 5 – 11 and Fig. 12 (i), in which it can be observed that the predicted values are less dispersed than the linear regression techniques, but are more spread out than the univariate imputation methods that lead to the best results. Although the Euclidean distance is considered in this study, various distance criteria can be utilised when implementing  $k$ -NN.

Support vector machines for regression (SVR) are also analysed, in which two kernel functions are employed (linear and RBF kernels). The SVR with linear kernel function model results in conjunction with the mean imputation results lead to the worst imputation results. Similarly to linear regression techniques, the predictions of the extreme values do not correspond with the values observed and, furthermore, the prediction range of these values are extremely high (see Fig 12 (j)). However, SVR with RBF kernel yields better results, in which the predicted values present less dispersion and are more similar to the observed values (see Fig 12 (k)), much like the  $k$ -NN imputation results (see Tables 5 - 11). Hence, the possibility of utilising various kernel functions makes the technique easily adaptable, and thus works accurately when dealing with either linear or non-linear relationships. It is also robust to outliers. However, poor performance may occur when treating noisy data, feature scaling is required, and both the kernel function and the tuning parameters need to be optimally selected. In addition, SVR is not convenient when the analysed sample is large, it is difficult to interpret, and its computational cost is high.

Furthermore, neural networks (NNs) with one, two, and three hidden layers are implemented. The three models yield similar results, although the execution time increases significantly when an additional hidden layer is employed. In addition, as expressed in Fig. 12 (l) – Fig. 12 (n), the predicted values do not correspond with the observed ones. One of the causes that leads to this inaccuracy may be the amount of data needed to train the model, the more data utilised the better the performance of the NN. In addition, the structure of the network is highly complex to define, they are susceptible to over-fitting, and their performances are unexplained. However, they capture both linear and non-linear relationships.

The most accurate results of the multivariate imputation techniques are obtained when the vector autoregressive (VAR) models are implemented. The results yield analogous results to ARIMA models (see Tables 5 – 11). This is a consequence of the algorithms implemented in both techniques, as VAR models are a generalisation of the autoregressive models, which are implemented within the ARIMA models. Hence, as indicated in Fig. 12 (o), the predicted values are remarkably similar to the observed records. A significant advantage of VAR models is that

750 they can impute various incomplete values of an instance by implementing the same model. Thus, if the instance contains three predictors with incomplete values at time  $t$ , they can be imputed by implementing the VAR method once only. Also, it is easy to implement and presents high accuracy when short-term data is considered. However, the analysed sample needs to be stationary and the order of the model needs to be optimally selected. It also presents high computation cost in comparison with other analysed methods.

755 To conclude with the models inquiry, decision tree regressors and ensemble methods are analysed as a whole, as they present similar results (expressed in Fig. 12 (p) – Fig. 12 (t)). Results are nearly equal when imputing incomplete values of the main engine fuel flow rate (with a MSE between 0.003 and 0.004 (tn/hr)<sup>2</sup>, see Table 7). Bagged tree ( $k$ -NN) yields better results when imputing missing values contained in the main engine fuel flow rate (with a RMSE of 0.037 tn/hr and a MedAE of 0.027 tn/hr, see Table 6) parameters. Conversely, decision tree regressor  
760 model imputation results are slightly better than the ensemble methods when imputing missing values from the scavenging air pressure of the scavenge air receiver (with a RMSE of 0.08 bars and a MedAE of 0.04 bars, see Table 11) is being analysed. Also, when Tables 5 – 11 are observed, it can be perceived that decision tree regressors and ensemble methods present results analogous to other non-linear regression methods analysed (SVR with RBF kernel and  $k$ -NN). Hence, these techniques may be applied if the linear relationship between the response and the predictors  
765 is not clearly identified. Additionally, decision tree regressors are easy to interpret, require less data pre-processing than other analysed models, can handle missing data, present low execution time, and are unlikely to over-fit. However, they are likely to present sub-optimal imputation performance when dealing with continuous data, are not accurate if instances with incomplete values to be imputed are not similar to instances that are utilised for training the model, and are also likely to be unstable, as only one tree is modelled and, thus, small adjustments in the training data  
770 may alter the partitions completely. To solve this last drawback, ensemble methods are implemented. Although the interpretability of the model is reduced, and the computational cost is increased.

A summary of both advantages and disadvantages identified for each imputation method implemented are summarised in Table 12. As expressed throughout this study, univariate imputation methods lead to better results than multivariate imputation methods, as large amounts of data are required to train the model every time a missing value  
775 is imputed. Furthermore, multivariate imputation methods cannot be performed if the analysed parameter does not present predictors or the instances to be utilised include predictors with missing values. Also, their computational cost is high, and thus they are not appropriate when the imputation is implemented in real-time. The only multivariate imputation techniques that lead to accurate results are the VAR models, which are a generalisation of the autoregressive models. This is due to the fact that these types of models are highly accurate when short-term data is  
780 considered. It is also easy to interpret and all parameters included in the model are considered endogenous. Thus, all incomplete values of an instance can be imputed by implementing the model once. However, these models can only

be utilised if data is stationary and, also, the execution cost may be higher than other imputation methods analysed, although it is not significant enough to exclude it for real-time frameworks. In consideration of the univariate imputation methods, all the techniques yield accurate results with the exception of mean imputation, which is not recommended due to their limitations, which are the distortion of the parameter distribution and the disruption of the relationship between features. STL decomposition and Holt Winters' seasonal method also yield accurate results, although they can only be applied when the time series present trend and seasonality. Furthermore, the estimation of the seasonal period is highly complex. For this reason, although the execution time is higher than the two methods previously mentioned, ARIMA models are considered the most appropriate univariate imputation techniques when dealing with short-term time series data in real-time, as they present higher accuracy when imputing incomplete values in short-term data and they are applicable to nearly all types of time series. However, as all the analysed methods, they also present some limitations, such as the necessity of the data to be stationary and the selection of the orders of the models.

Hence, after analysing all the implemented imputation techniques, it is suggested the utilisation of VAR when there are various missing values corresponding to different parameters that are correlated in the same instance and ARIMA models otherwise, as these are the models identified as the most accurate when short-term time series sensor data needs to be treated.

**Table 5.** Imputation results of the main engine rotational speed parameter.

|                         | Execution Time (s) | MSE ((r/min) <sup>2</sup> ) | MSLE (log((r/min) <sup>2</sup> )) | RMSE (r/min) | MAPE (%) | MedAE (r/min) | Max. Error (r/min) |
|-------------------------|--------------------|-----------------------------|-----------------------------------|--------------|----------|---------------|--------------------|
| Mean imputation         | 0.0002             | 11.49                       | 0.00011                           | 2.39         | 2.355    | 1.920         | 6.652              |
| STL decomposition       | 0.0234             | 0.066                       | 0.00001                           | 0.118        | 0.115    | 0.077         | 2.643              |
| Holt Winters            | 0.1384             | 0.071                       | 0.00001                           | 0.129        | 0.126    | 0.081         | 2.581              |
| ARIMA                   | 2.8614             | 0.063                       | 0.00001                           | 0.118        | 0.116    | 0.076         | 2.404              |
| PLS regression          | 0.0059             | 9.162                       | 0.00084                           | 1.989        | 1.952    | 1.731         | 9.746              |
| Ridge regression        | 0.0037             | 8.342                       | 0.00076                           | 1.920        | 1.882    | 1.718         | 9.455              |
| LASSO regression        | 0.0964             | 8.875                       | 0.00081                           | 1.976        | 1.939    | 1.854         | 9.604              |
| ElasticNet regression   | 0.0879             | 8.935                       | 0.00082                           | 1.981        | 1.944    | 1.855         | 9.570              |
| k-Nearest Neighbors     | 0.0025             | 4.666                       | 0.00041                           | 1.198        | 1.151    | 0.236         | 8.520              |
| SVR (linear kernel)     | 0.0401             | 10.215                      | 0.00094                           | 2.164        | 2.129    | 1.852         | 8.515              |
| SVR (RBF Kernel)        | 0.0268             | 4.799                       | 0.00043                           | 1.226        | 1.178    | 0.185         | 6.960              |
| NN (1 hidden layer)     | 53.3214            | 6.274                       | 0.00057                           | 1.636        | 1.593    | 1.193         | 7.107              |
| NN (2 hidden layers)    | 55.5707            | 5.913                       | 0.00054                           | 1.553        | 1.509    | 0.934         | 6.887              |
| NN (3 hidden layers)    | 57.6319            | 6.004                       | 0.00054                           | 1.562        | 1.518    | 0.737         | 6.826              |
| Vector autoregression   | 2.9989             | 0.085                       | 0.00001                           | 0.144        | 0.140    | 0.084         | 2.557              |
| Decision tree regressor | 0.0040             | 5.292                       | 0.00047                           | 1.143        | 1.095    | 0.140         | 9.170              |
| Bagged tree (SVR)       | 0.1115             | 4.770                       | 0.00042                           | 1.221        | 1.173    | 0.201         | 7.033              |
| Bagged tree (k-NN)      | 0.0208             | 4.494                       | 0.00040                           | 1.147        | 1.099    | 0.162         | 8.367              |
| Random forest           | 0.2710             | 4.621                       | 0.00041                           | 1.129        | 1.081    | 0.179         | 8.425              |
| AdaBoost                | 0.0259             | 4.020                       | 0.00036                           | 1.125        | 1.084    | 0.330         | 9.203              |

**Table 6.** Imputation results of the main engine power parameter.

|                         | Execution Time (s) | MSE (kW <sup>2</sup> ) | MSLE (log(kW <sup>2</sup> )) | RMSE (kW) | MAPE (%) | MedAE (kW) | Max. Error (kW) |
|-------------------------|--------------------|------------------------|------------------------------|-----------|----------|------------|-----------------|
| Mean imputation         | 0.0002             | 282912.4               | 0.0155                       | 423.433   | 10.541   | 419.265    | 938.840         |
| STL decomposition       | 0.0237             | 1808.440               | 0.0001                       | 29.691    | 0.688    | 23.877     | 330.164         |
| Holt Winters            | 0.1862             | 1867.622               | 0.0001                       | 30.660    | 0.716    | 24.158     | 329.228         |
| ARIMA                   | 1.5618             | 1520.140               | 0.0001                       | 27.123    | 0.631    | 21.868     | 303.016         |
| PLS regression          | 0.0066             | 165252.430             | 0.0084                       | 272.632   | 6.609    | 182.381    | 2109.975        |
| Ridge regression        | 0.0040             | 165309.421             | 0.0085                       | 273.729   | 6.637    | 183.178    | 2069.081        |
| LASSO regression        | 0.0849             | 176189.410             | 0.0092                       | 297.426   | 7.256    | 233.327    | 1986.912        |
| ElasticNet regression   | 0.0693             | 178432.169             | 0.0093                       | 299.607   | 7.311    | 242.007    | 1986.023        |
| k-Nearest Neighbors     | 0.0029             | 78476.090              | 0.0039                       | 189.779   | 4.394    | 111.675    | 1061.520        |
| SVR (linear kernel)     | 0.0439             | 418935.348             | 0.0195                       | 370.683   | 9.193    | 183.248    | 2469.608        |
| SVR (RBF Kernel)        | 0.0446             | 70917.784              | 0.0035                       | 176.527   | 4.052    | 89.021     | 919.772         |
| NN (1 hidden layer)     | 63.6674            | 128537.8               | 0.0070                       | 271.869   | 6.556    | 226.968    | 1436.062        |
| NN (2 hidden layers)    | 63.8124            | 112474.7               | 0.0059                       | 250.320   | 5.988    | 161.291    | 1224.988        |
| NN (3 hidden layers)    | 64.0917            | 115563.2               | 0.0061                       | 250.641   | 5.993    | 141.166    | 1172.232        |
| Vector autoregression   | 3.0802             | 2562.996               | 0.0001                       | 36.648    | 0.858    | 27.429     | 290.651         |
| Decision tree regressor | 0.0033             | 77802.296              | 0.0037                       | 171.422   | 3.885    | 72.340     | 1339.880        |
| Bagged tree (SVR)       | 0.1985             | 71858.359              | 0.0035                       | 178.552   | 4.107    | 89.405     | 925.058         |
| Bagged tree (k-NN)      | 0.0210             | 68086.261              | 0.0033                       | 167.098   | 3.819    | 96.821     | 1041.015        |
| Random forest           | 0.2512             | 61853.362              | 0.0030                       | 161.962   | 3.688    | 88.909     | 994.745         |
| AdaBoost                | 0.0217             | 60843.155              | 0.0030                       | 160.461   | 3.692    | 97.181     | 940.996         |

**Table 7.** Imputation results of the main engine fuel flow rate parameter.

|                         | Execution Time (s) | MSE ((tn/hr) <sup>2</sup> ) | MSLE (log((tn/hr) <sup>2</sup> )) | RMSE (tn/hr) | MAPE (%) | MedAE (tn/hr) | Max. Error (tn/hr) |
|-------------------------|--------------------|-----------------------------|-----------------------------------|--------------|----------|---------------|--------------------|
| Mean imputation         | 0.0002             | 0.009                       | 0.0025                            | 0.075        | 9.169    | 0.066         | 0.191              |
| STL decomposition       | 0.0234             | 0.001                       | 0.0002                            | 0.021        | 2.487    | 0.017         | 0.118              |
| Holt Winters            | 0.1988             | 0.001                       | 0.0002                            | 0.020        | 2.358    | 0.015         | 0.132              |
| ARIMA                   | 3.8901             | 0.001                       | 0.0002                            | 0.020        | 2.355    | 0.015         | 0.132              |
| PLS regression          | 0.0067             | 0.005                       | 0.0014                            | 0.052        | 6.239    | 0.037         | 0.330              |
| Ridge regression        | 0.0040             | 0.005                       | 0.0014                            | 0.052        | 6.188    | 0.037         | 0.330              |
| LASSO regression        | 0.0877             | 0.005                       | 0.0015                            | 0.055        | 6.561    | 0.042         | 0.329              |
| ElasticNet regression   | 0.0753             | 0.005                       | 0.0015                            | 0.055        | 6.589    | 0.042         | 0.328              |
| k-Nearest Neighbors     | 0.0028             | 0.003                       | 0.0009                            | 0.041        | 4.690    | 0.030         | 0.208              |
| SVR (linear kernel)     | 0.0432             | 0.010                       | 0.0026                            | 0.060        | 7.175    | 0.034         | 0.463              |
| SVR (RBF Kernel)        | 0.0501             | 0.003                       | 0.0008                            | 0.038        | 4.379    | 0.027         | 0.194              |
| NN (1 hidden layer)     | 41.4862            | 0.004                       | 0.0011                            | 0.047        | 5.610    | 0.036         | 0.213              |
| NN (2 hidden layers)    | 42.4137            | 0.004                       | 0.0011                            | 0.047        | 5.534    | 0.034         | 0.197              |
| NN (3 hidden layers)    | 43.2885            | 0.004                       | 0.0011                            | 0.047        | 5.561    | 0.034         | 0.191              |
| Vector autoregression   | 3.4904             | 0.001                       | 0.0004                            | 0.028        | 3.300    | 0.022         | 0.142              |
| Decision tree regressor | 0.0035             | 0.004                       | 0.0011                            | 0.043        | 4.971    | 0.030         | 0.280              |
| Bagged tree (SVR)       | 0.2094             | 0.003                       | 0.0008                            | 0.038        | 4.458    | 0.028         | 0.194              |
| Bagged tree (k-NN)      | 0.0208             | 0.003                       | 0.0008                            | 0.037        | 4.266    | 0.027         | 0.219              |
| Random forest           | 0.2638             | 0.003                       | 0.0008                            | 0.038        | 4.401    | 0.028         | 0.227              |

|          |        |       |        |       |       |       |       |
|----------|--------|-------|--------|-------|-------|-------|-------|
| AdaBoost | 0.0299 | 0.003 | 0.0008 | 0.039 | 4.579 | 0.028 | 0.192 |
|----------|--------|-------|--------|-------|-------|-------|-------|

**Table 8.** Imputation results of the inlet pressure parameter of the lubrication oil system.

|                   | Execution Time (s) | MSE (bars <sup>2</sup> ) | MSLE (log(bars <sup>2</sup> )) | RMSE (bars) | MAPE (%) | MedAE (bars) | Max. Error (bar) |
|-------------------|--------------------|--------------------------|--------------------------------|-------------|----------|--------------|------------------|
| Mean imputation   | 0.0002             | 0.00095                  | 0.000084                       | 0.021       | 0.911    | 0.008        | 0.061            |
| STL decomposition | 0.0236             | 0.00002                  | 0.000001                       | 0.002       | 0.103    | 0.001        | 0.013            |
| Holt Winters      | 0.2137             | 0.00001                  | 0.000001                       | 0.003       | 0.110    | 0.002        | 0.014            |
| ARIMA             | 5.3972             | 0.00002                  | 0.000001                       | 0.002       | 0.104    | 0.001        | 0.013            |

**Table 9.** Imputation results of the inlet pressure parameter of the jacket cooling water system.

|                   | Execution Time (s) | MSE (bars <sup>2</sup> ) | MSLE (log(bars <sup>2</sup> )) | RMSE (bars) | MAPE (%) | MedAE (bars) | Max. Error (bars) |
|-------------------|--------------------|--------------------------|--------------------------------|-------------|----------|--------------|-------------------|
| Mean imputation   | 0.0002             | 0.0005                   | 0.00002                        | 0.017       | 0.481    | 0.017        | 0.066             |
| STL decomposition | 0.0239             | 0.0006                   | 0.00003                        | 0.019       | 0.537    | 0.016        | 0.080             |
| Holt Winters      | 0.1656             | 0.0005                   | 0.00002                        | 0.017       | 0.491    | 0.014        | 0.070             |
| ARIMA             | 4.3112             | 0.0004                   | 0.00002                        | 0.017       | 0.468    | 0.014        | 0.066             |

**Table 10.** Imputation results of the turbine lubricating oil inlet pressure parameter of the turbocharger.

|                   | Execution Time (s) | MSE (bars <sup>2</sup> ) | MSLE (log(bars <sup>2</sup> )) | RMSE (bars) | MAPE (%) | MedAE (bars) | Max. Error (bars) |
|-------------------|--------------------|--------------------------|--------------------------------|-------------|----------|--------------|-------------------|
| Mean imputation   | 0.0002             | 0.026                    | 0.0026                         | 0.135       | 6.231    | 0.111        | 0.412             |
| STL decomposition | 0.0231             | 0.030                    | 0.0029                         | 0.142       | 6.506    | 0.126        | 0.498             |
| Holt Winters      | 0.1847             | 0.028                    | 0.0027                         | 0.139       | 6.360    | 0.131        | 0.438             |
| ARIMA             | 3.5027             | 0.026                    | 0.0025                         | 0.134       | 6.168    | 0.114        | 0.412             |

**Table 11.** Imputation results of the scavenging air pressure of the scavenge air receiver.

|                         | Execution Time (s) | MSE (bars <sup>2</sup> ) | MSLE (log(bars <sup>2</sup> )) | RMSE (bars) | MAPE (%) | MedAE (bars) | Max. Error (bars) |
|-------------------------|--------------------|--------------------------|--------------------------------|-------------|----------|--------------|-------------------|
| Mean imputation         | 0.0002             | 0.047                    | 0.0121                         | 0.17        | 19.97    | 0.18         | 0.41              |
| STL decomposition       | 0.0232             | 0.001                    | 0.0001                         | 0.01        | 1.53     | 0.01         | 0.18              |
| Holt Winters            | 0.2338             | 0.001                    | 0.0002                         | 0.02        | 1.55     | 0.01         | 0.19              |
| ARIMA                   | 1.6972             | 0.001                    | 0.0002                         | 0.01        | 1.50     | 0.01         | 0.18              |
| PLS regression          | 0.0068             | 0.037                    | 0.0085                         | 0.13        | 13.88    | 0.08         | 0.88              |
| Ridge regression        | 0.0047             | 0.036                    | 0.0085                         | 0.13        | 13.88    | 0.08         | 0.86              |
| LASSO regression        | 0.0775             | 0.037                    | 0.0086                         | 0.13        | 14.19    | 0.08         | 0.87              |
| ElasticNet regression   | 0.0803             | 0.037                    | 0.0088                         | 0.13        | 14.44    | 0.09         | 0.86              |
| k-Nearest Neighbors     | 0.0028             | 0.018                    | 0.0042                         | 0.09        | 9.62     | 0.06         | 0.47              |
| SVR (linear kernel)     | 0.0418             | 0.068                    | 0.0149                         | 0.15        | 17.10    | 0.08         | 1.02              |
| SVR (RBF Kernel)        | 0.0501             | 0.015                    | 0.0035                         | 0.09        | 8.66     | 0.05         | 0.35              |
| NN (1 hidden layer)     | 72.371             | 0.027                    | 0.0067                         | 0.12        | 13.47    | 0.09         | 0.65              |
| NN (2 hidden layers)    | 74.0086            | 0.023                    | 0.0057                         | 0.11        | 12.29    | 0.08         | 0.54              |
| NN (3 hidden layers)    | 75.7735            | 0.022                    | 0.0054                         | 0.11        | 11.91    | 0.07         | 0.50              |
| Vector autoregression   | 3.3160             | 0.001                    | 0.0002                         | 0.02        | 1.91     | 0.01         | 0.19              |
| Decision tree regressor | 0.0035             | 0.016                    | 0.0036                         | 0.08        | 7.92     | 0.04         | 0.54              |
| Bagged tree (SVR)       | 0.2176             | 0.015                    | 0.0036                         | 0.09        | 8.78     | 0.05         | 0.35              |

|                    |        |       |        |      |      |      |      |
|--------------------|--------|-------|--------|------|------|------|------|
| Bagged tree (k-NN) | 0.0205 | 0.015 | 0.0035 | 0.08 | 8.36 | 0.05 | 0.43 |
| Random forest      | 0.2711 | 0.014 | 0.0031 | 0.08 | 7.88 | 0.05 | 0.42 |
| AdaBoost           | 0.0217 | 0.014 | 0.0034 | 0.08 | 8.64 | 0.06 | 0.41 |

805

**Table 12.** Advantages and disadvantages of the implemented imputation techniques.

| Technique   | Advantages  | Disadvantages  |
|---|---|--|
| Mean imputation   | <ul style="list-style-type: none"> <li>• Easy to interpret and implement.</li> <li>• The execution time is low.</li> </ul>  | <ul style="list-style-type: none"> <li>• Distortion of the parameter distribution.</li> <li>• Disruption of the relationship between features.</li> <li>• Bias of the mean estimates when the nature of the incomplete values are either MAR or NMAR.</li> </ul>   |
| STL decomposition   | <ul style="list-style-type: none"> <li>• Robust to outliers.</li> <li>• Trend smoothness can be regulated.</li> <li>• Execution time is low.</li> <li>• Easy to interpret.</li> <li>• Any seasonality type can be considered, and the seasonal component can be adjusted over time.</li> </ul>  | <ul style="list-style-type: none"> <li>• Possibility to be applied only when the time series presents trend and seasonality.</li> <li>• The definition of the seasonal period is required.</li> </ul>  |
| Exponential smoothing methods   | <ul style="list-style-type: none"> <li>• Easy to interpret and implement.</li> <li>• Recent observations are considered more significant than earlier observations.</li> <li>• Various exponential smoothing methods can be applied based on the characteristics of the time series.</li> </ul> | <ul style="list-style-type: none"> <li>• The type of exponential smoothing method to be utilised needs to be identified.</li> <li>• Different parameters need to be optimally selected.</li> </ul>   |
| Autoregressive integrated moving average (ARIMA) models   | <ul style="list-style-type: none"> <li>• Present higher accuracy when imputing incomplete values in short-term data.</li> <li>• Applicable to nearly all types of time series.</li> </ul>   | <ul style="list-style-type: none"> <li>• Only captures linear relationships.</li> <li>• Need more data than other univariate imputation methods analysed.</li> <li>• Differencing is required if the data is not stationary. Then, the order of differencing needs to be specified.</li> <li>• The orders of the autoregression and the moving average model need to be optimally selected.</li> <li>• The computational cost is high in comparison with the other univariate imputation methods analysed, and thus its execution time is also large.</li> </ul> |
| Linear regression methods (PLS regression and penalised models (LASSO, Ridge, and ElasticNet regression)) | <ul style="list-style-type: none"> <li>• Easy to interpret.</li> <li>• Low execution time.</li> <li>• Regularisation models avoid overfitting.</li> <li>• Great performance when dealing with a linear relationship.</li> </ul>   | <ul style="list-style-type: none"> <li>• Sensitive to outliers.</li> <li>• Only captures linear relationships between the response and the predictors.</li> <li>• Risk when extrapolating.</li> <li>• Large amount of data required every time an incomplete value needs to be imputed.</li> <li>• Penalised parameters (penalised models) and the number of components to be utilised (PLS) need to be optimally selected.</li> </ul>   |
| <i>k</i> -Nearest Neighbors   | <ul style="list-style-type: none"> <li>• Different distance criteria can be implemented.</li> <li>• Easy to interpret.</li> </ul>   | <ul style="list-style-type: none"> <li>• Performance degradation when the sample considered is large or dimensions are high.</li> <li>• Feature scaling is needed.</li> </ul>  |

|   |  |   |
|---|--|---|
|   | <ul style="list-style-type: none"> <li>Weights can be added to the estimated distances hinging on the closeness of the records.</li> </ul>   | <ul style="list-style-type: none"> <li>Sensitive to outliers and noisy data.</li> <li>The number of neighbours required to impute the incomplete value needs to be optimally selected.</li> </ul>   |
| SVR (with linear and RBF kernels)   | <ul style="list-style-type: none"> <li>Captures both linear and non-linear relationships.</li> <li>Easily adaptable.</li> <li>Robust to outliers.</li> <li>Various kernel functions can be utilised.</li> </ul>  | <ul style="list-style-type: none"> <li>Poor performance when the sample contains noise.</li> <li>Inconvenient when the sample is large.</li> <li>Both the kernel function and the tuning parameters need to be optimally selected.</li> <li>Feature scaling is required.</li> <li>Difficult to interpret.</li> <li>High computational cost.</li> </ul>                              |
| Neural Networks (NN) (with 1, 2, and 3 hidden layers)   | <ul style="list-style-type: none"> <li>Capture both linear and non-linear relationships.</li> </ul>  | <ul style="list-style-type: none"> <li>Requires large amounts of data to train the model.</li> <li>Complexity in the network structure definition.</li> <li>Susceptible to over-fitting.</li> <li>High computational cost.</li> <li>Unexplained performance.</li> </ul>   |
| Vector autoregressive (VAR) models  | <ul style="list-style-type: none"> <li>Easy to interpret.</li> <li>High accuracy when dealing with short-term data.</li> <li>All parameters included in the model are considered endogenous.</li> <li>All incomplete values of an instance can be imputed by implementing the model once.</li> </ul> | <ul style="list-style-type: none"> <li>Can only be utilised if data is stationary.</li> <li>High computational cost.</li> <li>The order of the model needs to be optimally selected.</li> </ul>   |
| Decision tree regressors  | <ul style="list-style-type: none"> <li>Apply feature selection intrinsically.</li> <li>Require less data pre-processing.</li> <li>Easy to interpret.</li> <li>Handle incomplete values.</li> <li>Low execution time.</li> <li>Unlikely to over-fit.</li> </ul>                                       | <ul style="list-style-type: none"> <li>Likely to present sub-optimal imputation performance when continuous data is considered.</li> <li>Imputations are not accurate if instances with incomplete values are not similar to instances utilised for training the model.</li> <li>Instability.</li> </ul>  |
| Ensemble methods (Bagged trees (with $k$ -NN and SVR regressors), Random forests, and AdaBoost) | <ul style="list-style-type: none"> <li>Apply feature selection intrinsically.</li> <li>Require less data pre-processing.</li> <li>Handle incomplete values.</li> <li>Unlikely to over-fit.</li> </ul>  | <ul style="list-style-type: none"> <li>Complex to interpret.</li> <li>Higher computational cost than decision tree regressors.</li> <li>Likely to present sub-optimal imputation performance when continuous data is considered.</li> <li>Imputations are not accurate if instances with incomplete values are not similar to instances utilised for training the model.</li> </ul> |

## 5. Conclusions

The maritime industry currently offers state-of-the-art maintenance and inspection processes, an example of which is Condition-Based Maintenance (CBM), in which a large number of sensors are installed alongside the most critical components and around the environment where the assets are operating to implement CBM effectively. However, despite the undeniable benefits of their implementation in the maritime industry, several challenges, which includes dealing with missing values, need to be tackled.

Data imputation is a crucial pre-processing step, the aim of which is the estimation of the identified missing values, as, if they are not treated, the results derived from data analysis may be unreliable and inaccurate, leading to bias in further steps. Thus, poor models are obtained, which are then used in decision-making processes that assist energy efficient operations of marine machinery.

It is for these reasons that a critical literature review was implemented, in which the latest studies regarding data imputation were analysed to identify their possible utilisations and limitations in short-term sensor data of marine systems.

In addition, a methodology that provided a comparative study of widely used machine learning and time series forecasting algorithms was performed to assess not only their accuracy to impute incomplete values but also to evaluate their ability to impute these values in real-time. Accordingly, this study provided an extensive analysis of the advantages, disadvantages, and limitations of each implemented technique. The proposed methodology was holistic, and thus included identification of transient states and data preparation. Data preparation was divided into the data transformation and the correlation analysis steps. Based on the correlation results, both univariate and multivariate imputation methods were implemented if the parameter to be analysed presented at least one parameter. Otherwise, only the univariate imputation methods were applied. To evaluate the performance of the suggested models, missing values were generated completely at random and were imputed iteratively by considering time series cross-validation. For each iteration seven metrics were estimated (execution time, MSE, MSLE, RMSE, MAPE, MedAE and Max Error). Subsequently, once all the values were imputed, the mean of each metric was estimated to assess the overall performance of each model. Thus, the most appropriate imputation method could be determined.

To highlight the implementation of the proposed methodology, a case study on a total of 7 parameters including the main engine rotational speed, main engine power, main engine fuel flow rate, inlet pressure of the lubrication oil system, inlet pressure of the jacket water cooling system, turbine lubricating oil inlet pressure of the turbocharger, and scavenging air pressure of the scavenge air receiver) obtained from sensors that are installed on the marine machinery systems of a cargo vessel was utilised.

The comparative study concluded that autoregressive models yielded the best results when dealing with short-term sensor data. Hence, the utilisation of VAR models were suggested when there were various missing values corresponding to different parameters that were correlated in the same instance and ARIMA models otherwise, as these were the models identified as the most accurate when dealing with short-term time series data. However, their limitations cannot be dismissed. For instance, if the sample utilised to impute the incomplete values is not stationary, then their estimates may be biased. The proposed methodology can also be implemented as part of a framework to determine which algorithm is the most appropriate based on the characteristics of the data and its context, for example

when large gaps of missing values need to be treated and the data does not present either trend or seasonality or when  
845 either long-term or short term data are analysed.

Hence, data imputation is a data preparation practise that has gained popularity over the last few years due to its importance when dealing with IIoT sensor data. Thus, further research needs to be addressed. Some future work guidelines considered are described hereunder.

- 1) A comprehensive comparative methodology of deep learning models as imputation techniques.
- 850 2) Development of a monitoring and alerting tool to prevent the failure of the sensors.
- 3) The implementation of a real-time imputation technique in a holistic predictive framework to assist real-time data-driven decision-making strategies for energy efficient operations of marine machinery.

## References

- Aheleroff S., Xu X., Lu Y., Aristizabal M., Velásquez J. P., Joa B., Valencia Y., 2020. IoT-enabled smart appliances  
855 under industry 4.0: A case study. *Advanced Engineering Informatics* 43, pp. 1-14, URL <https://doi.org/10.1016/j.aei.2020.101043>.
- Azimi I., Pahikkala T., Rahmani A. M., Niela-Vilén H., Axelin A., Liljeberg P., 2019. Missing data resilient decision-making for healthcare IoT through personalization: A case study on maternal health. *Future Generation Computer Systems* 96, pp. 297-308, URL <https://doi.org/10.1016/j.future.2019.02.015>.
- 860 Balakrishnan S. M., Sangaiah A. K., 2018. Chapter 6 – aspect oriented modeling of missing data imputation for Internet of Things (IoT) based healthcare infrastructure. *Intelligent Data-Centric*, pp 135-145, URL <https://doi.org/10.1016/B978-0-12-813314-9.00006-2>.
- Bashir F., Wei H., 2018. Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing* 276, pp. 23-30, URL  
865 <https://doi.org/10.1016/j.neucom.2017.03.097>.
- Beck M. W., Bokde N., Asencio-Cortés G., Kulat K., 2018. R Package imputeTestbench to Compare Imputation Methods for Univariate Time Series. *The R Journal* 10, 218-233, URL <https://journal.r-project.org/archive/2018/RJ-2018-024/RJ-2018-024.pdf>.
- Bokde N., Beck M. W., Martínez Álvarez F., Kulat K., 2018. A novel imputation methodology for time series based  
870 on pattern sequence forecasting. *Pattern Recognition Letters* 116, pp. 88-96, URL <https://doi.org/10.1016/j.patrec.2018.09.020>.
- Cheliotis M., Gkerekos C., Lazakis I., Theotokatos G., 2019. A novel data condition and performance hybrid imputation method for energy efficient operations of marine systems. *Ocean Engineering* 188, pp. 1- 14. URL <https://doi.org/10.1016/j.oceaneng.2019.106220>.

- 875 Chivers B. D., Wallbank J., Cole S. J., Sebek O., Stanley S., Fry M., Leontidis G., 2020. Imputation of missing sub-hourly precipitation data in a large sensor network: A machine learning approach. *Journal of Hydrology* 588, pp. 1-12, URL <https://doi.org/10.1016/j.jhydrol.2020.125126>.
- Chong A., Lam K. P., Xu W., Karaguzel O. T., Mo Y., 2016. Imputation of missing values in building sensor data. *Building Performance Modeling Conference*, pp. 1-9.
- 880 Fekade B., Maksymyuk T., Kyryk M., Jo M., 2018. Probabilistic recovery of incomplete sensed data in IoT. *IEEE Internet of Things Journal* 5, pp. 2282-2292, URL <https://ieeexplore.ieee.org/document/7987674>.
- Fortuin V., Baranchuk D., Rätsch G., Mandt S., 2020. GP-VAE: Deep Probabilistic Time Series Imputation. *International Conference on Artificial Intelligence and Statistics* 23, pp. 1-17, URL <https://arxiv.org/abs/1907.04155>.
- 885 Gkerekos C., Lazakis I., Theotokatos G., 2019. Machine learning models for predicting ship main engine Fuel Oil Consumption: A comparative study. *Ocean Engineering* 188, pp. 1-14, URL <https://doi.org/10.1016/j.oceaneng.2019.106282>.
- Guo Z., Wan Y., Ye H., 2019. A data imputation method for multivariate time series based on generative adversarial network. *Neurocomputing* 360, pp. 185-197. URL <https://doi.org/10.1016/j.neucom.2019.06.007>.
- 890 Hadeed S. J., O'Rourke M. K., Burgess J. L., Harris R. B., Canales R. A., 2020. Imputation methods for addressing missing data in short-term monitoring of air pollutants. *Science of the Total Environment* 730, pp. 1-7, URL <https://doi.org/10.1016/j.scitotenv.2020.139140>.
- Hegde H., Shimpi N., Panny A., Glurich I., Christie P., Acharya A., 2019. MICE vs PPCA: Missing data imputation in healthcare. *Informatics in Medicine Unlocked* 17, pp. 1-8, URL <https://doi.org/10.1016/j.imu.2019.100275>.
- 895 Hyndman R. J., Athanasopoulos G., 2020. *Forecasting: principles and practice*. 3rd ed. Melbourne: Otexts, URL <https://otexts.com/fpp3/>.
- IMO, 2020. Greenhouse Gas Emissions. Retrieved from <http://www.imo.org/en/OurWork/Environment/PollutionPrevention/AirPollution/Pages/GHG-Emissions.aspx>
- 900 Izonin I., Kryvinska N., Tkachenko R., Zub K., 2019. An approach towards missing data recovery within IoT smart system. *Procedia Computer Science* 155, pp. 11-18, URL <https://doi.org/10.1016/j.procs.2019.08.006>.
- Kotu V., Deshpande B., 2019. Chapter 12 - time series forecasting. *Data Science*. 2nd ed. Burlington: Morgan Kaufmann, URL <https://doi.org/10.1016/B978-0-12-814761-0.00012-5>.
- 905 Kuhn M., Johnson K., 2016. *Applied Predictive Modeling*. 5th ed. New York: Springer Science + Business Media LLC, pp. 27-58.

- Lazakis I., Dikis K., Michala A. L., Theotokatos G., 2016. Advanced ship systems condition monitoring for enhanced inspection, maintenance and decision making in ship operations. *Transportation Research Procedia* 14, pp. 1679-1688, URL <https://doi.org/10.1016/j.trpro.2016.05.133>.
- 910 Lazakis I., Gkerekos C., Theotokatos G., 2018. Investigating an SVM-driven, one-class approach to estimating ship system condition. *Ships and Offshore Structures* vol. 14, pp. 432-441, URL <https://doi.org/10.1080/17445302.2018.1500189>.
- Lazakis I., Raptodimos Y., Varelas T., 2018. Predicting ship machinery system condition through analytical reliability tools and artificial neural networks. *Ocean Engineering* vol. 152, pp. 404-415, URL
- 915 <https://doi.org/10.1016/j.oceaneng.2017.11.017>.
- Liu Y., Dillon T., Yu W., Rahayu W., Mosafa F., 2020. Missing value imputation for Industrial IoT sensor data with large gaps. *IEEE Internet of Things Journal* (Early Access), URL <https://ieeexplore.ieee.org/abstract/document/8976165>.
- Luo Y., Cai X., Zhang Y., Xu J., Yuan X., 2018. Multivariate Time Series Imputation with Generative Adversarial Networks. *Advances in Neural Information Processing Systems* 31, pp. 1596-1607, URL <http://papers.nips.cc/paper/7432-multivariate-time-series-imputation-with-generative-adversarial-networks.pdf>.
- MAN Diesel & Turbo, 2010. 98-50 MC/MC-C-TII type engines. Engine selection guide. 1st ed. Copenhagen: MAN Diesel & Turbo SE.
- 925 Noor N. M., Abdullah M. M. A. B., Yahaya A. S., Ramli N. A., 2014. Comparison of linear interpolation method and mean method to replace the missing values in environmental data set. *Materials Science* 803, pp. 278-281, URL <https://www.scientific.net/MSF.803.278>.
- Pedregosa F. et al., 2011. Scikit-learn: Machine Learning in Python. *JMLR* 12, pp. 2825-2830, URL <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>.
- 930 Pratama I., Permanasari A. E., Ardiyanto I., Indrayani R., 2016. A review of missing values handling methods on time-series data. *International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 1-6, URL <https://ieeexplore.ieee.org/document/7858189>.
- Priya Stella Mary I., Arockiam L., 2017. Imputing the missing data in IoT based on the spatial and temporal correlation. *IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, pp. 1-4,
- 935 URL <https://ieeexplore.ieee.org/document/8249990>.
- Raptodimos Y., Lazakis I., 2018. Using artificial neural network self-organising map for data clustering of marine engine condition monitoring applications. *Ships and Offshore Structures*, vol. 13, pp. 649-656, URL <https://doi.org/10.1080/17445302.2018.1443694>.

- Raptodimos Y., Lazakis I., 2019. Application of NARX neural network for predicting marine engine performance parameters. Ships and Offshore Structures, vol. 15, pp. 443-452, URL <https://doi.org/10.1080/17445302.2019.1661619>.
- 940
- Seabold S., Perktold J., 2010. Statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference, pp. 92-96. URL <https://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf>.
- 945
- Siegel A., Morgan C., 1996. Statistics and data analysis. An introduction. 2nd ed. Toronto: John Wiley & Sons, Inc., pp. 12-100.
- Taylor D. A., 1996. Introduction to Marine Engineering. 2nd ed. Oxford: Elseiver Ltd., pp. 8-53.
- Wärtsilä, 2020. Combustion engine for power generation: introduction, viewed 27 June 2020, URL <https://www.wartsila.com/energy/learn-more/technical-comparisons/>.
- 950
- Yoon J., Jordon J., vand der Schaar M., 2018. GAIN: Missing Data Imputation using Generative Adversarial Nets. 2018 International Conference of Machine Learning, URL <https://arxiv.org/abs/1806.02920>.