Conducting Vessel Data Imputation Method Selection Based on Dataset Characteristics

Tirana Noor Fatyanosa[1] **Neni Alya Firdausanti**[1] **Luis Japa**[1]
Israel Mendonça[1] **Putu Hangga Nan Prayoga**[2] **Masayoshi Aritsugi**[1]

[1]Kumamoto University, Kumamoto 860-8555, Japan
[2]MTI Co., Ltd., Tokyo 100-0005, Japan

E-mail: fatyanosa@dbms.cs.kumamoto-u.ac.jp,
nenialya@dbms.cs.kumamoto-u.ac.jp, japa-luis@dbms.cs.kumamoto-u.ac.jp,
israel@cs.kumamoto-u.ac.jp, putu.hangga@monohakobi.com,
aritsugi@cs.kumamoto-u.ac.jp

Abstract. Time series datasets collected from marine sensors inevitably undergo missing data problems. This cause unreliable sensor data to assist the decision-making process. Many methods are offered to impute missing values. However, selecting the best imputation method is not a trivial task, as it usually requires domain expertise and several trial-and-error iterations. Furthermore, when imputations are carried out in a careless way, it generates a high error factor that can lead stakeholders to wrong assumptions. This paper provides a systematic approach that is able to extract characteristics of underlying data and, based on it, recommends the less error-prone imputation method. We evaluate our proposed method using nine real-world vessel datasets. In total, we generated 3859 data samples consisting of 17 inputs and 1 target feature. Experimental results show that the proposed approach is capable of obtaining a weighted F1-Score of 92.6%. Additionally, when compared with the application of careless selected imputation methods, our work is able to gain up to 86% on the average imputation score, with the worst case gain being of 5%. We empirically demonstrate that the proposed approach is efficient when selecting the best imputation methods.

1. Introduction

Time series are important in practical applications as time series data must be handled correctly to avoid erroneous and biased results, eventually leading to a flawed decision-making process [1]. However, missing values are inevitable in time series caused by unexpected events, such as malfunctioning sensors or missing signals. Ignoring the observation with missing values is an easy alternative. There is no big issue when there are only a few observations with missing values. However, a significant amount of information is lost when several observations have missing values. Additionally, it reduces the data's statistical power and effectiveness. Therefore, reliable imputation methods are required to address the missing data problem [2].

In a time series dataset, missing block patterns might be highly random and diverse. Additionally, the dataset may have quite various properties depending on the length and number of series, the frequency of repetitions (seasonality) within a series, and the connection between series. There may be a complete continuous block of entries missing from a time series or several different time series. Depending on the size of the block, its position to other missing blocks, patterns within a series, and correlation (if any) with other series in the dataset, the signals from the rest of the dataset that are most relevant for imputing a missing block will vary. Interpolation with close neighbors may be helpful if only one entry is missing.

Repeated patterns within the series and trends from connected series may be helpful if a range of values from a particular time series is lacking. Only patterns within a series will be helpful if the same time range is missing from many series [3].

In the context of data analysis, data imputation is a pre-processing stage aiming to estimate pinpointed missing values in order to avoid the under-utilization of data. Hence, if missing values are not tackled, the results obtained may be unreliable and inaccurate, leading to biases in further phases due to inadequate models implemented in the decision-making process [4]. Although time series' data imputation is a well-researched subject, the choice of the best missing data handling technique is still a challenging issue. It relies on several variables, and trade-offs between various elements and there is no golden standard that can be used in every case and have optimal results; instead, selecting the optimal method depends on a combination of interrelated factors [5]. Moreover, it is presumed that the dataset's characteristics may have some bearing on the algorithm. This hypothesis drove us to model the relationship between the dataset's characteristics and the effectiveness of each missing value imputation method to select the best method accordingly.

Our main contributions are described below:

- We study the datasets and propose 17 features obtained from their characteristics.
- We provide 3859 dataset samples that can be used directly to study the effects of different imputation methods.
- We assess our proposed approach using nine real-world vessel datasets. We investigate eight imputation methods. As a result, interpolation and MissForest [6] became the most dominant imputation methods.

The remainder of this paper is structured as follows: Section 2 provides related work. Section 3 shows the proposed methodology. Section 4 reflects the experimental setup. Section 5 presents results and discussion of the implementation of the proposed methodology. Lastly, Section 6 concludes the overall results.

2. Related Work

The major approaches employed to handle missing values are categorized into three groups: deletion methods, donor-based methods, and model-based methods. Deletion methods are approaches that remove the rows that contain missing values [7]. These approaches are a common strategy if the percentage of missing data is less than 5% [8]. Contrarily, if the percentage of missing data is more than 5% then donor-based imputation or model-based imputation are recommended. In donor-based imputation the missing values in a record (row) are filled with data from another record (or records) with similar characteristics. Some examples of donor based imputation methods are: hot-deck imputation and k-Nearest Neighbours (kNN) imputation. Regarding model-based imputation, a predictive model is designed for each target variable in the data set that contains missing values. Usually, this model is fitted on the available data and then used to impute the missing values.

Another possible way in which imputation methods can be divided is into the two following groups: single imputation methods and multiple imputation methods. In single imputation methods the missing values are imputed just once, without defining a model for the partially missing data. Some methods included in this category are mean, mode, median, interpolation and k-Nearest Neighbour (kNN) imputation. Mean, mode, and median imputation are methods that use statistical values (average, mode and median, respectively) to impute missing values. Further, interpolation is an imputation technique that assumes a linear relationship between missing and non-missing values. Finally, kNN imputation finds the k-most similar observations by taking into account all of the other features and averages them to fill the missing values. Inside the family of kNN, [9] proposed GkNN, which is able to efficiently handle data with mixed-attributes.

In contrast to single imputation, multiple imputation techniques combine the results of several imputed datasets to arrive to a final solution. Some examples of commonly used multiple imputation

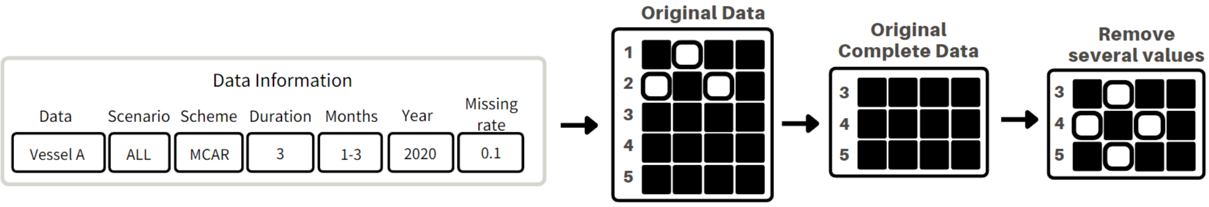


Figure 1. Data selection and preparation based on the data information.

methods are: MissForest, Multiple Imputation by Chained Equations (MICE), Multiple Imputation Using Denoising Autoencoders (MIDA) and Missing Data Imputation using Generative Adversarial Nets (GAIN). MissForest is a random forest-based iterative imputation method proposed in [6]. This method first imputes all the missing data using a single imputation method, such as mean or mode, and then fits a random forest on the observed part of the data in order to predict the missing values. MICE was developed in [10] and it fills the missing values through an iterative series of predictive models. Similar to MissForest, at the beginning the missing values are "initialized" using another imputation method. Common choices for this step are mean and mode. Then, at every iteration the MICE algorithm imputes the missing values individually by fitting a model on the rest of the data. The iterations continue until a user-defined condition is met. MIDA was developed in [11] and it is based on a type of artificial neural networks called Denoising autoencoders (DAEs). Autoencoders consist of two artificial neural networks: an encoder and a decoder. The encoder take some input and map it to an intermediate representation; then, the decoder maps this representation back to its original domain. The main assumption is that this intermediate representation captures the coordinates along the main factors of variation. GAIN, developed in [12], adapts the well-known Generative Adversarial Nets (GAN) framework to the data imputation domain. In this work the generator observes some components of a real data vector and imputes the missing ones. The discriminator then takes this completed vector and attempts to determine which components were actually observed and which were imputed.

3. Methodology

We study how different imputation methods behave for different scenarios. Our process consists of different steps explained as follows:

First, we define the data information shown in Table 1. For the vessel data information, we define two scenarios, ALL (all columns: 217) and IMP (important columns: 57), based on the importance of the columns referring to causalities for the use case of machinery predictive maintenance. For the scheme, we use three missing values mechanisms: Missing completely at random (MCAR), Missing not at random (MNAR), and Missing at random (MAR). We use the missing values settings from [13].

From the data information, we can obtain the subset of datasets. Then, we remove the missing rows to get the original complete data. Next, we synthetically remove some cells based on the scheme and missing rates defined in the data information. The visualization can be seen in Figure 1.

We then divided the process into the input and target generations. Input features are the measurement that is obtained easily from the dataset characteristics. To generate suitable features, we emphasize missing data-related features and follow some references [5, 14, 15]. The input features we used can be seen in Table 1. To obtain the features, we first artificially removed some values from the original complete data for particular data information. In total, we selected 17 input features from the data to test.

Target feature generation consists of three main steps: data imputation, best method selection, and generation. In the data imputation step, each data information will be imputed with eight imputation methods: interpolation, mean, mode, median, miceforest, KNN, GAIN, and MissForest. Each method will record the RMSE, MAE, and time. Then the process continued to the second step: best method selection. For each data information, the recorded RMSE, MAE, and time will be used to calculate the score using Equation 1. The best method for particular data information is selected based on the lowest score.

Table 1. Data information, input, and target features.

Category	Component	Description
Data Information	Data Scenario Scheme Duration Year Months Missing rate	The dataset name [ALL, IMP] [MCAR, MNAR, MAR] [3, 6, 12, 24] [2020,2021,2020-2021] [3: Jan-Mar, Apr-Jun, Jul-Sep, Oct-Dec, 6: Jan-Jun, Jul-Dec, 12: Jan-Dec, 24: Jan 2020-Dec 2021] [0.1, 0.2, 0.3, 0.4, 0.5, real]
Input Features	A: Missing rate B: Number of incomplete rows C: Percentage of incomplete rows D: Number of complete rows E: Percentage of complete rows F: Number of incomplete columns G: Percentage of incomplete columns H: Number of complete columns I: Percentage of complete columns J: Average missing cells per row K: Average missing cells per column L: Number of all null features M: Number of uniform features N: Number of rows O: Number of columns P: Number of missing cells Q: Number of maximum consecutive NaNs	The values were taken from the properties of the data.
Target Features	Best Method	The values were taken from the best imputation method for each data information.

Table 2. The statistics of the datasets.

Dataset	Number of samples	Number of columns
Pure Car Carrier (Vessel A)	171164	127
LPG Tanker (Vessel B)	165261	127
Ro-Ro Cargo Ship (Vessel C)	174898	127
Container Vessel (Vessel D)	173987	127
Oil Tanker (Vessel E)	160343	127
Ro-Ro Cargo Ship (Vessel F)	102138	127
LPG Tanker (Vessel G)	17391	127
Ro-Ro Cargo Ship (Vessel H)	175056	127
Oil Tanker (Vessel I)	17517	127

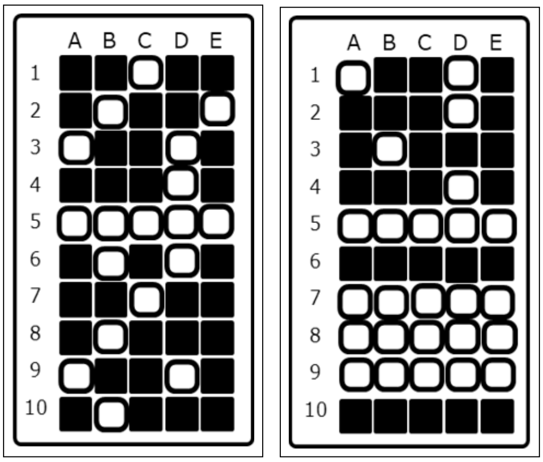
4. Experimental Setup

We applied the proposed method to the real vessel datasets. We use an IoT sensor time series dataset of commercial ocean-going vessels of various sizes and types. Datasets for all vessels were collected for two years periods with 6 minutes sampling frequency. Statistics of the datasets are outlined in Table 2. For more details, summary statistics of selected columns from Vessel A can be seen in Table A1 of Appendix A.

Focusing on the vessel data, we found two common missing patterns for vessel IoT data, as shown by Table 3. The two vessels in the table have distinctly different missing patterns, MCAR and MNAR, respectively (see Figure 2 for the illustration). If we process these missing data using only the deletion

Table 3. Characteristic of missing pattern for vessel data.

Characteristics	Vessel A	Vessel C
Built year	1999	2018
LoA (m)	199.94	179.94
Breadth (m)	32.2	27
DWT	21547	6100
Design draft (m)	8.75	6.5
Main Engine	Mitsubishi 8UEC60LS	MAN 9S50ME
Number of Main Engine Cylinders	8	9
Data collection frequency	every 6 minutes	every 6 minutes
Duration of collected data	2020-2021	2020-2021
Missing by Rows		
Total rows	171164	174898
Complete rows	0	45
Incomplete rows	171164	174853
Missing by Cell		
Total cells	16431744	21862250
Missing cells	1999295	764933
Avg missing cells (columns)	20826	6120
Avg missing cells (rows)	12	4
Common missing pattern	MCAR (Figure 2a)	MNAR (Figure 2b)



(a) MCAR

(b) MNAR

Figure 2. Illustration of missing pattern.

method, it is natural to have fewer data to perform advanced analytics. In some cases, leave 0% of the data available for modeling as shown by Figure 3.

We conducted two types of data preprocessing: data cleaning and data transformation. Data cleaning consists of removing unnecessary space, changing the empty string into NaN, removing empty columns, and removing columns with the same values in all rows. Data transformation consists of transforming all data into numerical data.

To measure the quality of the imputation methods, we utilized three metrics: mean absolute error (MAE), root mean square error (RMSE), and normalized execution time ($\bar{t}$). Those three measurements were combined into one (see Equation 1) to discover a more resounding conclusion and avoid misleading. The MAE and RMSE are the main metrics. The time metric is supplementary and not too important; therefore, we add a weight of 0.1. Upon calculating the metrics, we transform the data within the same range between 0 and 1 using the Min-Max Scaling.

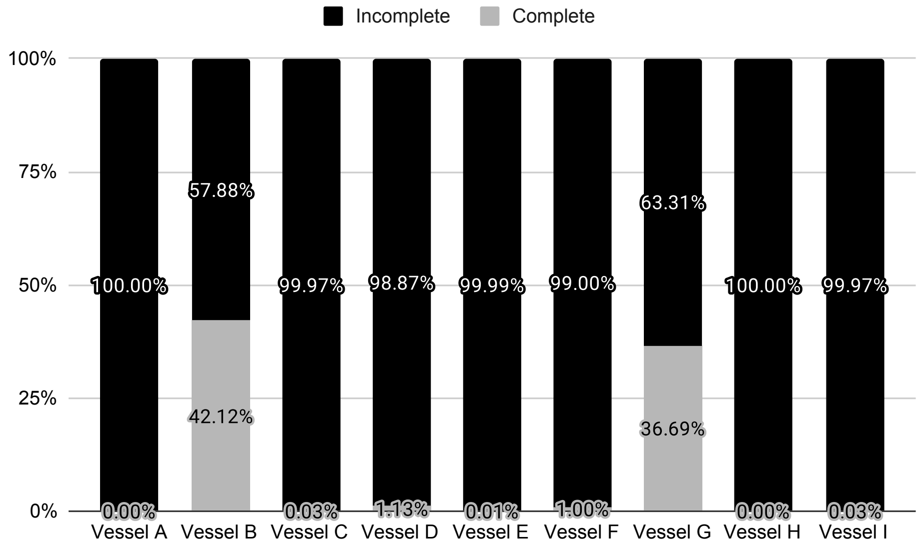


Figure 3. Percentage of leftover data for analysis when performing imputation by deletion method.

Table 4. Target class ratio of all imputation methods.

Method	Mean	Mode	Median	Interpolation	KNN	miceforest	GAIN	MissForest
Ratio	0 (0.00%)	0 (0.00%)	3 (0.08%)	1527 (39.57%)	30 (0.78%)	57 (1.48%)	0 (0.00%)	2242 (58.10%)

Note: The values in the table represent the total and percentage (in the bracket)

$$score = (RMSE + MAE) * (1 + (0.1 * \bar{t})) \tag{1}$$

where $\bar{t}$ is the Min-Max normalized execution time.

5. Results and Discussion

From the features generation, we extract 3859 dataset samples. All samples consist of 17 input features and 1 target feature. We begin our discussion by assessing the performance of the imputation methods. The first observation is that, in all cases, MissForest and interpolation dominated the overall best methods (Table 4).

Even though GAIN is considered the best-performing imputation method [16], it cannot outperform interpolation and MissForest. One possible explanation is that GAIN is ineffective in imputing time series [17]. Comparing the performance, we can conclude that the simple classical methods outperform advanced imputation methods.

We assume that the domination of interpolation and MissForest in performance is possible because of the nature of both methods that are appropriate for the time series dataset. For interpolation, when data of a column is missing at a particular time, it is natural to fill the missing data with a straight line between the two points of time. Meanwhile, MissForest has been proven to be a reliable imputation method for time series datasets with complex interactions and nonlinear relations [18, 19].

To investigate the difference between both methods more deeply, we had compared the number of maximum consecutive NaNs as shown in Figure 4. It is shown that most of the data samples that favor interpolation as the best method has lower numbers of maximum consecutive NaNs than the data samples of MissForest. It indicates that the missing gap is an important part of method selection. Interpolation works best with smaller gaps between consecutive records. When the gap is high, the prediction ability

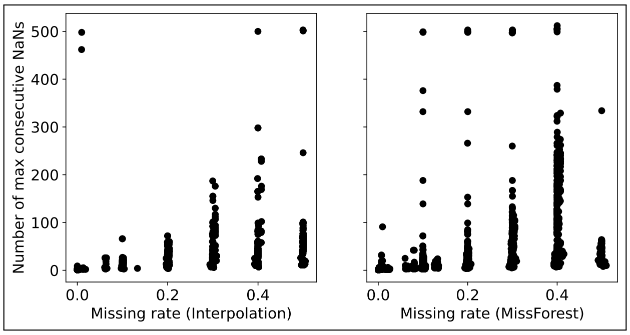


Figure 4. Number of max consecutive NaNs for all methods.

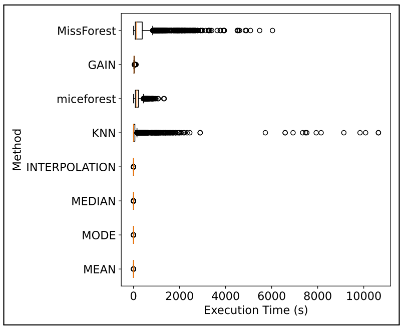


Figure 5. Execution time analysis.

becomes limited as the interpolation fits a gap between the last and next observation data [20]. Meanwhile, MissForest seems to perform well regardless of the gap size.

The boxplot of overall execution time (Figure 5) shows that although all the methods impute the same number of missing cells, the execution time is different. The boxplot demonstrates two conclusions: (1) All statistical methods, mean, mode, median, and interpolation, only need a lower time to impute all missing values, and (2) the other advanced methods need a higher execution time.

The XGBoost shows that the generated dataset can be successfully used to select the best imputation method, as the weighted F1-Scores is 92.6%. Moreover, our approach significantly improves the average imputation score. In a no-recommendation scenario, in which the same imputation method was used for all datasets, the obtained average scores per method were 0.115, 0.504, 0.671, 0.514, 0.127, 0.233, 0.295, and 0.101 for interpolation, mean, mode, median, miceforest, KNN, GAIN, and MissForest, respectively. However, with the employment of our approach, the average score was reduced to 0.096. The best theoretical possible value for this measurement, i.e. the average score in a perfect recommendation scenario, is 0.095. Our obtained value is notably close to this theoretical value, evidencing the effectiveness of our proposed method.

6. Conclusion and Future Remarks

In this paper, we present a systematic imputation method selection based on dataset characteristics. We conduct experiments using nine real-world vessel datasets. We obtained 3859 dataset samples with 17 input and 1 target features. Interpolation and MissForest were found to be the dominant imputation methods. The testing results showed that our approach was able to reach 92.6% of the weighted F1-Score using XGBoost. Compared with the results without our approach, the highest and lowest average imputation score improvement were 86% and 5%, respectively.

The current version of the algorithm only recommends the whole data. However, in some cases, it is better to make a mixed imputation, in which we perform imputation for subsets of the data using different methodologies. By incorporating domain knowledge, we will adapt the algorithm for future work to use a subset of columns instead of the whole data.

Acknowledgement

The authors wish to express their gratitude to MTI Co., Ltd. for providing the vessel datasets.

References

[1] Fekade B, Maksymyuk T, Kyryk M and Jo M 2018 *IEEE Internet of Things Journal* **5** 2282–2292

[2] Khan S I and Hoque A S M L 2020 *Journal of Big Data* **7** ISSN 2196-1115

[3] Bansal P, Deshpande P and Sarawagi S 2021 *CoRR* **abs/2103.01600** (*Preprint* `2103.01600`) URL `https://arxiv.org/abs/2103.01600`

[4] Velasco-Gallego C and Lazakis I 2021 *RINA Maritime Innovation and Emerging Technologies Online Conference 2021 Proceedings* (United Kingdom: Royal Institution of Naval Architects) royal Institution of Naval Architects Maritime Innovation and Emerging Technologies Online Conference 2021, RINA Maritime Innovation and Emerging Technologies Online Conference 2021 ; Conference date: 17-03-2021 Through 18-03-2021 URL `https://www.rina.org.uk/Maritime_innovation_2021.html`

[5] Liu Q and Hauswirth M 2020 *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* pp 0349–0358

[6] Stekhoven D J and Bühlmann P 2011 *Bioinformatics* **28** 112–118 ISSN 1367-4803 (*Preprint* `https://academic.oup.com/bioinformatics/article-pdf/28/1/112/583703/btr597.pdf`) URL `https://doi.org/10.1093/bioinformatics/btr597`

[7] Wothke W 2000 *Modeling longitudinal and multilevel data: Practical issues, applied approaches, and specific examples* (Lawrence Erlbaum Associates Publishers) pp 219–240

[8] Graham J W 2009 *Annual Review of Psychology* **60** 549–576 ISSN 0066-4308 URL `https://www.annualreviews.org/doi/10.1146/annurev.psych.58.110405.085530`

[9] Zhang S 2012 *Journal of Systems and Software* **85** 2541–2552 ISSN 0164-1212 URL `https://www.sciencedirect.com/science/article/pii/S0164121212001586`

[10] White I R, Royston P and Wood A M 2011 *Statistics in Medicine* **30** 377–399 ISSN 02776715 URL `https://onlinelibrary.wiley.com/doi/10.1002/sim.4067`

[11] Gondara L and Wang K 2018 *Advances in Knowledge Discovery and Data Mining* (Springer International Publishing) pp 260–272 URL `https://doi.org/10.1007%2F978-3-319-93040-4_21`

[12] Yoon J, Jordon J and van der Schaar M 2018 *Proceedings of the 35th International Conference on Machine Learning* (*Proceedings of Machine Learning Research* vol 80) ed Dy J and Krause A (PMLR) pp 5689–5698 URL `https://proceedings.mlr.press/v80/yoon18a.html`

[13] Muzellec B, Josse J, Boyer C and Cuturi M 2020 *Proceedings of the 37-th International Conference on Machine Learning* (Vienna, Austria: arXiv) URL `https://arxiv.org/abs/2002.03860`

[14] Zou Y, An A and Huang X 2005 *2005 IEEE International Conference on Granular Computing* vol 2 pp 728–733 Vol. 2

[15] Sim J, Lee J S and Kwon O 2015 *Mathematical Problems in Engineering* **2015** ISSN 15635147

[16] Awan S E, Bennamoun M, Sohel F, Sanfilippo F and Dwivedi G 2021 *Neurocomputing* **453** 164–171 ISSN 0925-2312 URL `https://www.sciencedirect.com/science/article/pii/S0925231221005282`

[17] Luo Y, Zhang Y, Cai X and Yuan X 2019 (International Joint Conferences on Artificial Intelligence Organization) pp 3094–3100 ISBN 978-0-9992411-4-1

[18] Zhang S, Gong L, Zeng Q, Li W, Xiao F and Lei J 2021 *Remote Sensing* **13**(12) ISSN 20724292

[19] Arriagada P, Karelovic B and Link O 2021 *Journal of Hydrology* **598** 126454 ISSN 0022-1694 URL `https://www.sciencedirect.com/science/article/pii/S0022169421005011`

[20] Adhikari D, Jiang W, Zhan J, Rawat D B, Aickelin U and Khorshidi H A 2022 *ACM Computing Surveys* ISSN 0360-0300

Appendix A.

Table A1. Summary statistics of selected columns from Vessel A.

Column	Mean	St.Dev.	Min.	25%	50%	75%	Max.
M/E RPM (rpm)	46.07	38.92	-120.00	0.00	59.00	85.00	97.00
M/E LOAD (%)	27.87	26.77	0.00	0.57	21.03	56.79	81.93
LOG SPEED (AVERAGE) (knot)	6.00	6.76	0.00	0.00	0.38	11.87	17.86
OG SPEED (AVERAGE) (knot)	6.64	6.48	0.00	0.88	2.32	12.34	20.00
M/E NO.1 CYL EXH GAS OUT T (C)	230.06	141.11	17.00	48.00	311.00	352.00	402.00
M/E NO.1 CYL SCAV AIR T (C)	54.41	6.64	31.00	51.00	54.00	58.00	72.00
M/E NO.1 CYL JACKET COOLING FRESH WATER OUT T (C)	77.67	10.06	32.00	76.00	81.00	83.00	94.00
M/E NO.1 CYL PCO OUT T (C)	47.55	6.13	21.00	42.00	48.00	53.00	60.00
M/E FO IN P (MPa)	0.53	0.10	0.00	0.51	0.54	0.57	0.72
M/E FO IN T (C)	98.99	24.47	23.00	98.00	106.00	113.00	139.00
M/E MAIN LO IN P (MPa)	0.26	0.09	0.00	0.26	0.27	0.31	0.33
M/E MAIN LO IN T (C)	44.95	3.81	24.00	43.00	44.00	47.00	63.00
M/E SCAV AIR IN P (MPa)	0.04	0.04	0.00	0.00	0.02	0.08	0.17
M/E SCAV AIR IN T (C)	41.95	4.81	25.00	39.00	43.00	45.00	62.00
M/E NO.1 T/C TACHOMETER (x100 rpm)	52.80	48.54	0.00	0.00	47.50	105.60	139.60
M/E NO.1 T/C EXH GAS IN T (C)	270.60	170.24	22.00	52.00	358.00	426.00	463.00
M/E NO.1 T/C EXH GAS OUT T (C)	234.81	146.52	18.00	43.00	333.00	357.00	401.00
ROLL (AVERAGE) (deg)	9.73	0.85	6.23	9.26	9.75	10.25	13.26
PITCH (AVERAGE) (deg)	0.96	0.24	-0.17	0.80	0.97	1.13	1.92
YAW (AVERAGE) (deg)	-4.57	102.80	-179.38	-95.95	-8.94	86.67	179.24
HEADING (deg)	198.00	95.96	0.00	132.06	197.37	283.25	360.00
RUDDER ANGLE (AVERAGE) (deg)	NaN	NaN	NaN	NaN	NaN	NaN	NaN
RELATIVE WIND DIRECTION (deg)	NaN	NaN	NaN	NaN	NaN	NaN	NaN
RELATIVE WIND (m/s)	NaN	NaN	NaN	NaN	NaN	NaN	NaN