Decision Science Letters 13 (2024) 197–210

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Machine learning models for condition-based maintenance with regular truncated signals

Tyler Ward[a], Kouroush Jenab[a*] and Jorge Ortega-Moody[a]

[a]*Department of Engineering & Technology Management, Morehead State University, Morehead, KY 40351, United States*

C H R O N I C L E

Article history:
Received: July 5, 2023
Received in revised format: August 21 , 2023
Accepted: September 30, 2023
Available online: September 30, 2023

Keywords:
Condition monitoring
Machine learning
Maintenance

A B S T R A C T

Condition-based maintenance (CBM) of industrial machines depends on the continuous, real-time monitoring of the machine's operational condition via smart sensors attached to different components on the machine. The problem of regularly spaced missing data, which can occur due to a variety of hardware or software issues, is one that is often overlooked in the literature surrounding CBM in industrial machines. Such missing data can cause issues in interpreting the true operational state of the machine, which can reduce the effectiveness of CBM processes. In this paper, we examine the capabilities of five data imputation techniques for handling this regular missing data and examine the impact these techniques have on machine learning (ML) classification algorithms for machine fault diagnosis. We examine the following techniques: simple mean imputation, mean imputation with outliers removed, best and worst-case imputation, and previous day imputation. Each of these methods is configured with the specific parameters that they will only consider data from the previous 24 hours, to ensure that the data is recent, and adequately represents the current status of the machine. The efficacy of each method at accurately reconstructing the missing data and the impact they have on ML classification is recorded in the results. The models are evaluated on a real-world dataset and are evaluated on a variety of common performance metrics.

1. Introduction

In an increasingly digitized industrial landscape, data-driven maintenance approaches have become central to optimizing machine performance and longevity. This increasing dependence on real-time data monitoring is demonstrated by the rapid adoption of predictive maintenance techniques such as condition-based maintenance (CBM). CBM is a maintenance strategy that relies on the real-time analysis of data received from smart sensors attached to different components of an industrial machine to determine its current operational state and to trigger maintenance activities only at the point of necessity, right before the failure point of the machine (Merkt, 2019; Loukopoulos et al., 2016). This approach, unlike corrective or preventive measures, aims to reduce costly downtime and repairs by avoiding unnecessary maintenance.

There are numerous ways that the effectiveness of CBM can be reduced, but perhaps one of the most common issues that affects the efficacy of CBM approaches is the presence of missing data in the training data (Merkt, 2019; Loukopoulos et al., 2016). Missing data can arise from a variety of factors, such as sensor malfunctions, network issues, or routine maintenance activities. Regardless of cause, gaps in the data used to inform the maintenance systems can lead to incorrect assumptions about a machine's condition and suboptimal decision-making (Loukopoulos et al., 2016; Osman et al., 2018; Du et al., 2020; Alabadla et al., 2022). Conventional techniques for handling missing data may not be sufficient and could compromise the dependability of predictive maintenance models like CBM.

* Corresponding author. Tel.: +1-606-783-9339 Fax: +1-606-783-5030
E-mail address: k.jenab@moreheadstate.edu (K. Jenab)

doi: 10.5267/dsl.2023.9.006

In Industry 4.0, there has been a rapid rise in the use of machine learning (ML) algorithms to improve data analytics and machine maintenance strategies (Osman et al., 2018; Alabadla et al., 2022). One relevant application of ML to the subject of this paper is data imputation, which is the process of replacing missing values with substitutes in an attempt to reconstruct the shape of the original data and preserve as much information as possible. In this paper, we evaluate several ML-based data imputation models specifically designed to address the issue of missing data that occurs in regular intervals in the context of CBM. The specific focus of this research is on missing data pertaining to temperature sensors in industrial machinery, which serve as important indicators for the health and performance of the machines (Hey et al., 2016).

The evaluated models utilize data from the previous 24-hour operational cycle. The models include simple mean imputation, mean imputation with outliers removed, best case imputation, worst case imputation, and previous day imputation. Each of these models draw on patterns and trends from the previously recorded cycles to estimate the most probable values that could fill the missing data points within the parameters of the respective model.

To empirically assess the performance of the imputation models, we artificially introduce missing data to the data taken from the temperature sensors. These gaps are designed to simulate the frequency of missing data commonly encountered in real-world scenarios. Once the gaps have been introduced, the models are trained on the incomplete data and evaluated based on their ability to reconstruct the shape of the original dataset, as well as how they impact the performance of ML-based fault diagnosis using the random forest (RF) ML classifier.

The goal of this study is to examine the effectiveness of different ML-based data imputation models, most of which are uncommon in the field of maintenance, to demonstrate their effectiveness for maintenance-related tasks, and promote further exploration into how these under-studied methods can be expanded for use in maintenance. We aim to enhance the robustness and reliability of CBM by refining these models for such applications. Through a comprehensive evaluation of these models, we hope to contribute to more accurate and dependable data-driven decision-making processes in industrial settings.

2. Literature Review

Successfully implementing multiple imputation techniques relies on understanding the nature of missing data. According to Du et al. (2020) and Osman et al. (2018), missing data can be broadly classified into three categories: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR), each having different characteristics and scenarios where they occur, such as in air pressure systems (Rafsunjani et al., 2019), control charts (Alwan et al., 2022), and sudden equipment failures. The presence of missing data in condition monitoring data can significantly reduce the accuracy and reliability of CBM models, which necessitates the need for exploration into the impact of different imputation models.

Traditional data imputation techniques, such as listwise and pairwise deletion, and simple methods, such as mean and model-based imputations, have been applied in various maintenance contexts (Hartini, 2017; Song et al., 2020; Appoh & Yunusa-Kaltungo, 2021; Martins et al., 2022; Song et al., 2022). Because of the simple and conventional nature of these methods, they may not be capable of adequately addressing complex challenges presented by missing data in CBM and may potentially impair the dependability of the maintenance models. More advanced imputation methods, such as multiple imputations and ML-based methods, have also been explored, and have demonstrated adaptability in diverse scenarios (Li & He, 2015; Zhang et al., 2022; Wang et al., 2023). However, the application of these methods to regularly truncated signals in CBM scenarios has not been extensively researched, leaving a considerable gap for exploration and refinement in the area of industrial maintenance.

The nature of the industry greatly influences the prevalence of certain imputation models. For instance, two methods that have found use in the medical industry that haven't made their way to the maintenance field are best-case and worst-case imputation, such as described in Pederson et al. (2017). In that study, the best-case and worst-case imputation were examined in the context of a patient's exposure to disease, with the best-case scenario being that after a follow-up with their doctor, the patient remained alive, and the worst-case scenario being that the patient died. These best-case and worst-case imputation models have shown their usefulness in identifying the impact of missing data (Jakobsen et al., 2017), but have also shown that they can lead to certain biased estimations (Barnes et al., 2010). As stated, the best-case and worst-case imputation techniques have been examined in the context of the medical industry, but in our review of the literature surrounding data imputation techniques for maintenance purposes, especially in CBM strategies, we found no evidence that these techniques have been explored in this way, especially not in the context of using them to aid CBM for processes that have regular truncated signals. Research into this application could be beneficial in determining the ability of such models to positively impact CBM strategies.

Another imputation technique that has been used in the medical institutions that could have major relevance to CBM strategies, is something that we will define as previous day imputation. This strategy could have particular relevance to CBM given the common time-series format of CBM datasets. In this strategy, which has been employed in studies such as Srimedha et al. (2022), missing values are directly substituted with whatever value was present at the same time 24-hours previously. Such a method has the potential to seamlessly integrate with the inherent structure of a CBM process and could potentially provide an adequate way to address regular truncated signals for CBM.

The current body of literature in this area provides an insight into the application of various imputation methods in diverse fields. However, there is a lack of exhaustive research focusing on the efficacy of these methods in addressing regularly spaced missing data, specifically in the context of CBM. The integration and evaluation of these methods, tailored to the unique needs and challenges of CBM with regularly truncated signals, remains largely unexplored, demonstrating the need for focused exploration and empirical assessments. This paper seeks to bridge the existing gaps in literature by providing empirical insights into the effectiveness of the selected imputation techniques in mitigating the impact of regularly spaced missing data on ML models for CBM.

3. Methodology

3.1. Machine Temperature System Failure Dataset

In this paper, we employ a dataset of temperature sensor readings from an internal component of a large, industrial machine. The dataset is designed primarily to aid research into anomaly detection methods within the sensor data of the machine. It is a publicly accessible dataset (Lavin & Ahmad, 2015) and has been widely referenced in existing literature (Adriana Mercioni & Holban, 2022). The dataset consists of samples over an 80-day period, and the data is presented in time-series format. There are multiple anomalies present in the data during the operational period described, with information provided about how each anomaly affected the condition of the machine. The timestamps for the anomalies were provided in the documentation of the dataset, allowing for detailed exploration into the temperature variations prior to and after the occurrence of an anomaly. The anomalies are varied in nature and range from a scheduled maintenance operation to a catastrophic failure of the machine. The description of the behavior of the anomalies is given by the authors of the dataset. There are four anomalous instances in the dataset. The first is indicated by a sharp spike towards higher temperature, the second is a planned shutdown of the machine for maintenance reasons. The third anomaly is difficult to detect, and under evaluation, was found to have led directly to the fourth anomaly, which was a catastrophic failure of the machine. Fig. 1 visually depicts the data, with the anomalies, and ranges of data points before and after the anomaly noted.

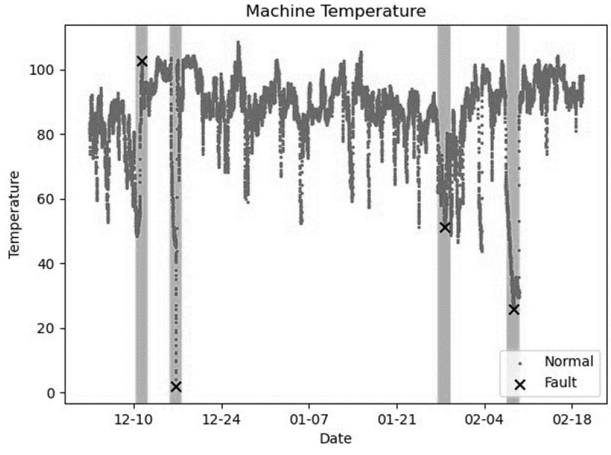

Fig. 1. Scatter plot of machine temperature sensor data.

3.2. Data Selection

The real-time monitoring of the condition of a machine is crucial to the proper implementation of a CBM strategy. This is because in industrial machines, which often have very complicated architecture with many intricate components, the operational loads and ambient conditions are subject to spontaneous fluctuations, which renders machine states highly dynamic. Given the importance of real-time condition monitoring data to the success of CBM strategies, this study prioritizes the most recent operational data to form the basis for our data imputation models. In industrial contexts, the relevance of condition data can degrade rapidly, making information beyond 24 hours in age potentially misrepresentative of the machine's current operation. The use of outdated data in decision-making processes can increase the likelihood of unexpected interruptions in the operation of the machine. In light of this, our methodology focuses on the evaluation of historical data that does not exceed a 24-hour window prior to any instance of missing data. The four distinct ranges that cause failures are distinguished from the normal functioning data in this new dataset. The timestamps at which each range starts and ends are saved in appropriately designated variables, and these ranges are stored in an array. The original data and the percentage of the data that is to be changed are passed as parameters into a function that replaces the existing data values with NaN values along with each index of this array of ranges. Nine distinct percentages are studied for the objectives of this work, ranging from 10% of data being missing to 90%.

The algorithm used first identifies values in the original data that align with timestamps in the range leading up to an anomaly. This subset is then processed to introduce NaN values in place of real data points, based on an MNAR scheme. The data with missing values added is stored in a separate data frame, preserving the original data so that it can be used for

validation purposes of the effectiveness of the imputation models at recreating it later in the experiment. The dataframe that contains the data with missing values is converted from a Pandas (The Pandas Development Team, 2020; McKinney, 2010) dataframe to a NumPy (Harris, 2020) array, which allows for easier manipulation.

The specific MNAR approach used ensures that the missing data is concentrated within predefined failure ranges, maintaining a form of regularity. Within these ranges, the data points are iteratively grouped into blocks, with the missingness percentage dictating the amount of data that is replaced with NaN values in each block. This provides a structured yet versatile way to represent missingness in data leading up to an anomaly in the condition of a machine, allowing for a comprehensive evaluation of the various imputation models. Algorithm 1 describes the pseudocode of this process.

Algorithm 1 Replace Data with MNAR values

function replaceWithMNAR(*df*, *percentage*, *failureRanges*)
    for each (*start*, *end*) in *failureRanges* **do**
        dataRange ← *df*[(*df.index* ≥ *start*) ∧ (*df.index* ≤ *endl*)]
        data ← *dataRange.values*
        for *x* ← 0 to *len*(*data*) with step 50 **do**
            for *y* ← 0 to 50 **do**
                if (*len*(*data*) – *x*) < 50 **then**
                    if $\frac{y}{(len(data)-x)} \leq \frac{percentage}{100}$ **then**
                        data[*x* + *y*] ← *NaN*
                    else
                        break
                    end if
                end if
                if $\frac{y}{50} \leq \frac{percentage}{100}$ then
                    data[*x* + *y*] ← *NaN*
                else
                    break
                end if
            end for
        end for
        df.loc[*dataRange.index*, :] ← *data*
    end for
    return *df*
end function

3.3. Generating Missing Not at Random Data

In order to examine the impact that different imputation models have on accurately reconstructing regular truncated signals, we needed to simulate the appearance of missing data in the dataset at regular time intervals. To do this, we designed an algorithm to introduce MNAR data to our existing dataset. The utilization of this missing data approach is meant to simulate real-world events that may occur in the context of CBM systems. In smart sensor networks, data might not go missing randomly, instead going missing due to some underlying mechanism or failure, meaning that the missing data would occur in regular intervals.

Given the emphasis of this paper on analyzing the impact of different imputation models on identifying anomalies in the condition of a machine, it is important to ensure that the missing data introduced to the dataset is extracted from the range of data that is in proximity to an anomaly. To facilitate this, the dataset was downsampled from an initial 22,695 samples to 4,572 samples, with the samples in this new version of the dataset being the anomalies themselves and the range of data leading up to and after the anomalies, and data from the 24-hour period before and after the faulty range. In this downsampled version of the data, the number of days containing observed data was reduced from 80 to 20.

3.4. General Algorithm Information

Each algorithm to be evaluated accepts the input data with missing data and returns a new dataframe with the imputed values now substituted in place of the missing ones. The input data is expected to have time-indexed rows and multiple columns that represent different features of sensor readings. The algorithms iterate through each column and data points in the input data. The time complexity of each algorithm is O(*nm*), where *n* is the number of rows and *m* is the number of columns in the input data. Because each data point in the input data is iterated through, the overall time complexity for each algorithm is linear with respect to the size of the data.

3.5. Mean-Based Machine Learning Approaches to Data Imputation

In the presence of missing data, the simple mean imputation algorithm calculates the mean of the data within the last 24 hours and replaces the missing value with this mean. This algorithm leverages temporal proximity, assuming that the underlying data points exhibit some level of temporal consistency, which makes the mean a reasonable choice for imputation.

Algorithm 2 Simple Mean Imputation

```
function simpleMeanImputation(missingData)
        copy missingData
        for each column in missingData do
                for each (index, value) in missingData do
                        if valueIsMissing then
                                calculate the mean of the values in the previous 24 hours
                                replace the missing value, store in missingDataCopy
                        end if
                end for
        end for
        return missingDataCopy
end function
```

Fig. 2 shows a side-by-side comparison of the missing data and the reconstructed data after simple mean imputation.

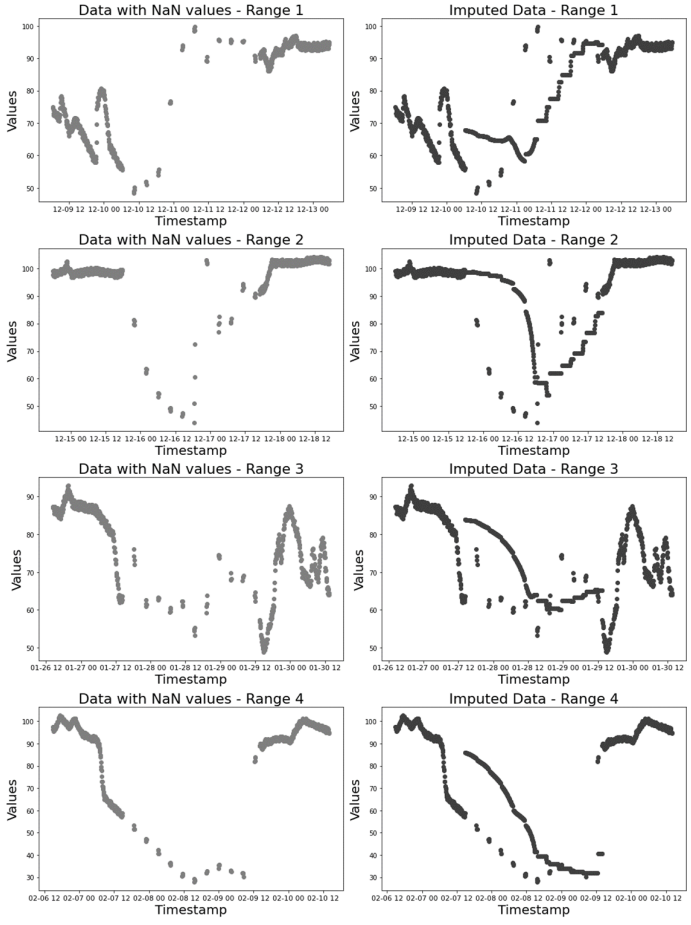

Fig. 4. Comparison of the missing data to the reconstructed data after simple mean imputation.

From looking at the reconstructed data, it is apparent that visually the simple mean imputation technique is not adequate for recreating the shape of the dataset. To attempt to improve this approach, outlier removal was introduced to the imputation method. This was deemed necessary because outliers can affect the accuracy of the data imputation (Akouemo & Povinelli, 2017). This paper uses the Local Outlier Factor (LOF) method for outlier removal. LOF is a highly effective approach for identifying outliers in a dataset and is particularly useful for anomaly detection in ML-based CBM strategies. LOF is based

on a density-based approach for outlier detection, allowing for the discernment of local outliers that may not be discernible using global thresholding methods. The LOF method computes the local density deviation of an object with respect to its neighbors to identify regions of similar density, and points that have a substantially lower density compared to their neighbors are considered outliers (Jha, 2019; Emir, 2023). It commences by computing the local reachability density of each data point, subsequently comparing these densities to discern outliers. A high LOF score implies the presence of an outlier, with the score reflecting the degree of abnormality. The algorithm operates by assessing the ratio of the average local density of a data point's k nearest neighbors to its own local density (Schmidt, 2020). Objects with a significantly lower density than their neighbors are deemed outliers. Here, k is a user-defined parameter representing the number of neighbors to be considered for calculating the local density. To achieve the most accurate results, LOF was employed before NaN samples were added to the dataset. From 4,572 samples in the original data, 15 outliers were identified and removed, leaving 4,557 samples remaining in the dataset, after which NaN samples were introduced, and the previously defined 24-hour mean imputation technique was employed. The reconstructed data using simple mean imputation after outlier removal using LOF compared to the missing data is shown in Fig. 5.

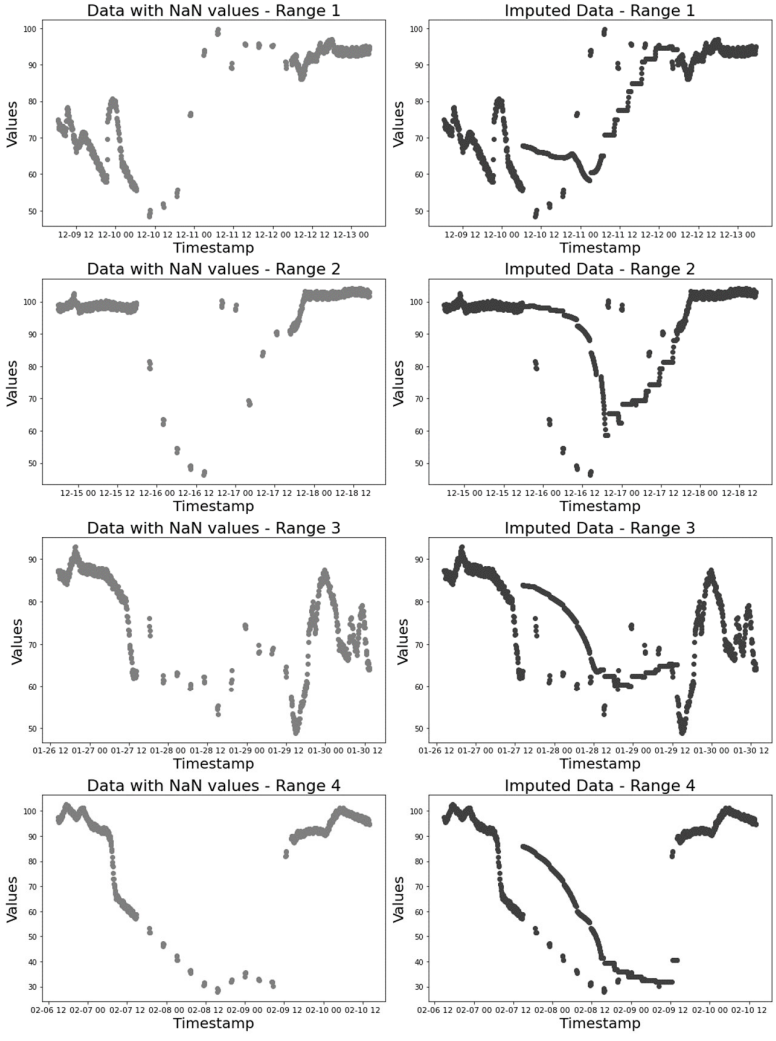

Fig. 5. Comparison of the missing data to the reconstructed data after mean imputation with outliers removed.

3.6. Best- and Worst- Case Approaches to Data Imputation

The best-case imputation algorithm offers a nuanced approach to imputing missing values in time-series data, leveraging both temporal locality and the intrinsic distribution of the data to make more informed imputations. It operates by finding the mean of the previous 24-hours of data, and then finding the value closest to this mean in the previous 24-hours of data. If there is missing data, then it is imputed based on this identified closest value. Unlike either of the mean imputation techniques already discussed, which could blur variations in the data, this strategy may better preserve the underlying data distribution and fluctuations. It could be particularly useful when the closest data point is more representative of the missing value than a generalized measure like this mean. In contrast, the worst-case imputation algorithm operates by finding the mean of the previous 24-hours of data, but instead of finding the closest value to the mean, it finds the furthest value from the mean, and uses that value in the imputation process. Like its base-case counterpart, worst-case imputation could

potentially provide better representation of the underlying system's dynamics where a conventional mean-based imputation may be inadequate. By choosing the furthest value from the mean within the fixed 24-hour window, this method captures extreme behavior that could be significant for certain types of analyses, such as anomaly detection. However, worst-case imputation by name has drawbacks. One such example is that it could introduce greater variance into the dataset and could potentially distort the distribution of whichever feature is being analyzed. The imputed values in this method are heavily influenced by outliers, which may not be appropriate for all types of time-series data. In summary, the worst-case imputation method may be particularly relevant for situations where it is necessary to capture the most extreme behaviors of a system, but it may not be universally applicable.

Algorithm 3 Best- or Worst-Case Imputation

```
function bestOrWorseCaseImputation(missingData, case)
        copy missingData
        for each column in missingData do
                for each (index, value) in missingData do
                        if valueIsMissing then
                                calculate the mean of the values in the previous 24 hours
                                if case = ”best” then
                                        find the closest value to the mean within the last 24 hours
                                        replace the missing value, store in missingDataCopy
                                end if
                                if case = ”worst” then
                                        find the furthest value from the mean within the last 24 hours
                                        replace the missing value, store in missingDataCopy
                                end if
                        end if
                end for
        end for
        return missingDataCopy
end function
```

Fig. 6 shows a side-by-side comparison of the missing data and the reconstructed data after best-case imputation.

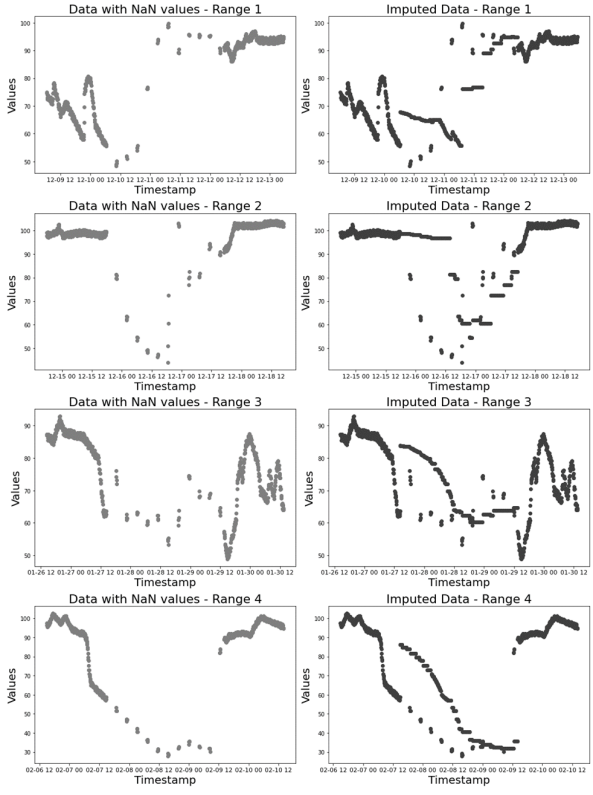

Fig. 6. Comparison of the missing data to the reconstructed data after best-case imputation

Fig. 7 shows a side-by-side comparison of the missing data and the reconstructed data after worst-case imputation.

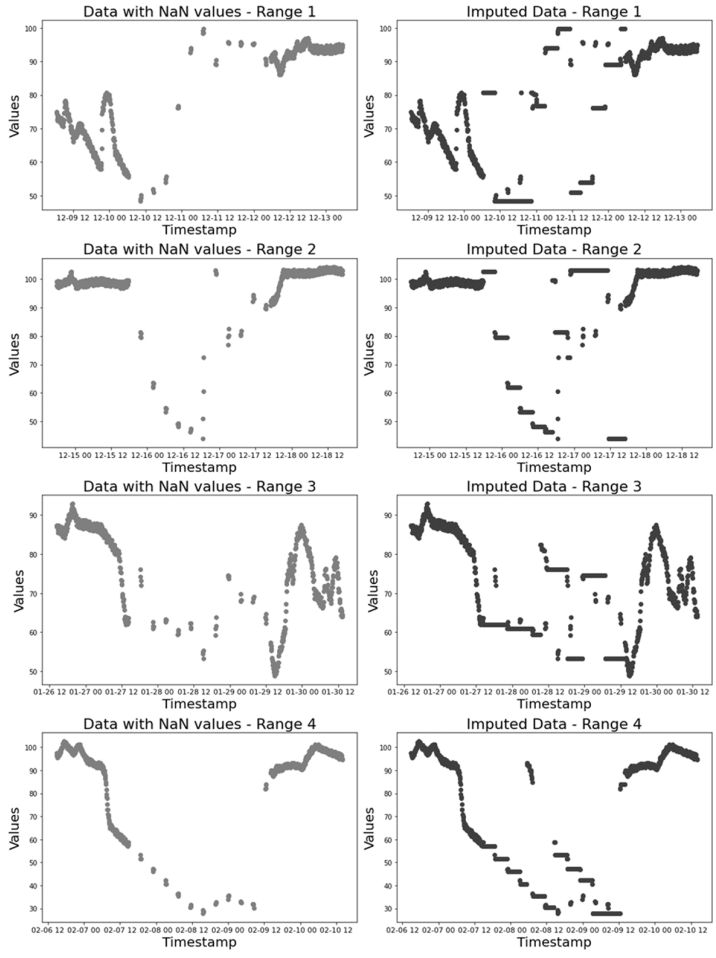

Fig. 7. Comparison of the missing data to the reconstructed data after worst-case imputation.

3.7. Previous Day Imputation

The previous data imputation algorithm aims to impute missing values by utilizing temporal relationships in the dataset. Specifically, it substitutes a missing value with data from the same time on a previous day. When such data is unavailable, the algorithm resorts to mean imputation. This is a hybrid approach to data imputation, leveraging temporal patterns for imputation while also falling back to mean-based imputation when required. This makes the algorithm versatile and more resilient to different kinds of missing data patterns. However, the algorithm is not without limitations, such as the assumption that the underlying process is periodic to some extent, which may not be true for all time-series data. In addition, if the dataset has significant shifts in behavior or is not well-represented by its mean, the mean imputation fallback might not be appropriate.

Algorithm 4 Previous Day Imputation

function simpleMeanImputation(*missingData*)
    copy *missingData*
    for each column in *missingData* **do**
        for each (*index*, *value*) in *missingData* **do**
            if *valueIsMissing* **then**
                determine the value at the same timestamp of the missing value on the previous day
                replace the missing value, store in *missingDataCopy*
            end if
        end for
    end for
    return *missingDataCopy*
end function

Fig. 8 shows a side-by-side comparison of the missing data and the reconstructed data after previous-day imputation.

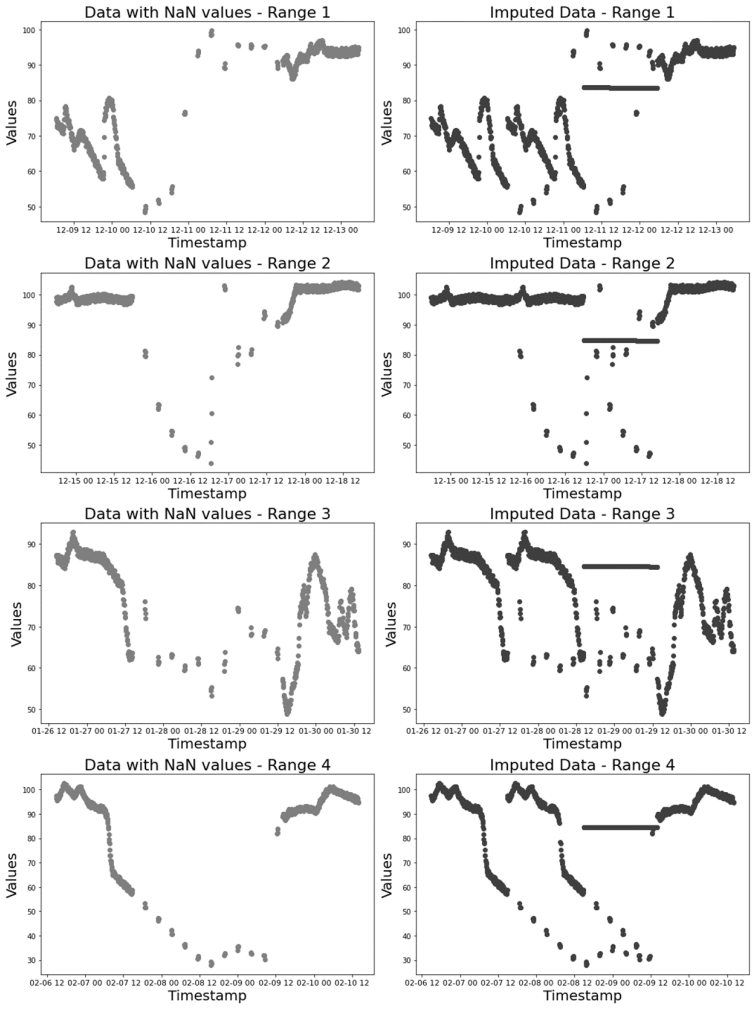

Fig. 8. Comparison of the missing data to the reconstructed data after previous-day imputation

3.8. Fault Classification on Imputed Data

Once the five imputation models were implemented, a supervised ML model was used to classify machine faults based on the imputed data. For the purposes of this paper, the RF classifier was used. This algorithm was selected because it commonly ranks as one of the most prevalent and accurate classifiers for maintenance-related tasks (Accorsi et al., 2017; Çınar et al., 2020). The RF approach relies on a collection of decision tree (DT) classifiers $\{h(x, \Theta_k), k=1, ...\}$, where Θ_k are the parameters of the tree. Each DT is constructed using a different bootstrap sample of the data, and during the tree-building process, at each node, a random subset of features is chosen to determine the best split. This randomness in selection of samples and features while constructing individual trees helps in achieving a diversified ensemble, which consequently reduces overfitting and variance. When an input is fed into the RF classifier for prediction, it traverses down each decision tree and reaches a final leaf node. The leaf node contains the predicted class, and the final classification is determined by aggregating the predictions from all trees in the forest. For classification tasks, a majority voting scheme is typically employed, where the class that receives the most votes by the individual trees is chosen as the final output class. RF holds several advantages over individual DTs. The most notable one is its resilience to overfitting due to the averaging process, which helps to cancel out the biases of the individual trees. Furthermore, it is robust to noise and can handle imbalanced datasets effectively. RF also provides feature importance scores, which are helpful for interpretability and understanding the significance of different input variables in making predictions.

4. Results

Table 1 shows the performance metrics of each of the proposed data imputation methods from ranges of missing data from 10% - 90%. These metrics measure the performance of the models in accurately reconstructing the original data. The performance metrics used are given by the following formulas:

1) Mean Absolute Error (MAE): the measure of errors between paired observations.

$$MAE = \frac{\sum_{i=1}^{n}|e_i|}{n} \tag{1}$$

2) Mean Absolute Percentage Error (MAPE): expresses accuracy as a ratio.

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{2}$$

3) Mean Squared Error (MSE): the average squared difference between the estimated values and the actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{3}$$

4) Root Mean Squared Error (RMSE): the square root of the MSE.

$$RMSE = \sqrt{MSE} \tag{4}$$

5) Coefficient of Determination (R^2): the proportion of the variation in the dependent variable that is predictable from the independent variable. The formula for R^2 relies on the residual sum of squares (SS_{res}) and the total sum of squares (SS_{tot}).

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{5}$$

Table 1
Evaluation metrics for data reconstruction

Missing Data %									
Metric	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
	Simple Mean Imputation								
MAE	0.90%	1.68%	2.49%	3.31%	4.17%	5.03%	7.73%	9.00%	10.50%
MAPE	1.50%	2.79%	4.17%	5.63%	7.25%	9.25%	11.96%	13.78%	15.92%
MSE	4.54%	8.67%	13.17%	17.82%	22.85%	28.07%	33.39%	39.55%	47.31%
RMSE	21.31%	29.44%	36.29%	42.22%	47.80%	52.98%	57.79%	62.89%	68.78%
R^2	95.46%	91.33%	86.83%	82.18%	77.15%	71.93%	66.61%	60.45%	52.69%
Simple Mean Imputation (Outliers Removed)									
MAE	1.13%	2.09%	3.09%	4.12%	5.15%	6.16%	7.24%	8.48%	10.05%
MAPE	1.46%	2.71%	4.04%	5.45%	6.83%	8.25%	9.71%	11.45%	13.51%
MSE	4.43%	8.31%	12.47%	16.70%	20.98%	25.33%	30.00%	36.08%	44.44%
RMSE	21.04%	28.83%	35.31%	40.86%	45.81%	50.33%	54.74%	60.06%	66.66%
R^2	95.57%	91.69%	87.53%	83.30%	79.02%	74.67%	70.04%	63.92%	55.56%
Best-Case Imputation									
MAE	0.90%	1.69%	2.50%	3.30%	4.16%	5.07%	7.64%	9.06%	10.58%
MAPE	1.50%	2.80%	4.19%	5.62%	7.26%	9.29%	11.87%	13.85%	16.02%
MSE	4.53%	8.73%	13.26%	17.77%	22.91%	28.42%	32.96%	40.58%	48.94%
RMSE	21.29%	29.54%	36.41%	42.15%	47.86%	53.31%	57.41%	63.71%	69.96%
R^2	95.47%	91.27%	86.74%	82.23%	77.09%	71.58%	67.04%	59.42%	51.06%
Worst-Case Imputation									
MAE	1.46%	2.67%	3.84%	4.91%	5.77%	6.79%	7.01%	7.51%	7.32%
MAPE	2.47%	4.48%	6.48%	8.35%	9.70%	12.01%	12.59%	12.90%	12.00%
MSE	17.30%	31.62%	45.82%	59.23%	70.14%	83.22%	37.60%	38.37%	32.91%
RMSE	41.59%	56.23%	67.69%	76.96%	83.75%	91.23%	61.32%	61.94%	57.37%
R^2	82.70%	68.38%	54.18%	40.77%	29.86%	16.78%	62.40%	61.63%	67.09%
Previous-Day Imputation									
MAE	1.49%	2.72%	3.91%	5.12%	6.40%	7.69%	11.83%	13.65%	15.70%
MAPE	2.66%	4.86%	7.37%	10.08%	13.07%	16.64%	21.38%	24.69%	28.25%
MSE	11.40%	20.91%	29.81%	39.12%	49.68%	60.99%	72.78%	84.67%	97.53%
RMSE	33.77%	45.72%	54.60%	62.54%	70.48%	78.09%	85.31%	92.02%	98.76%
R^2	88.60%	79.09%	70.19%	60.88%	50.32%	39.01%	27.22%	15.33%	2.47%

From these results, it is evident that as the percentage of missing data increases, the performance deteriorates for all models, as indicated by higher errors (MAE, MAPE, MSE, RMSE) and lower R^2 scores. This was expected behavior, as the more data is missing, the less original data the models have to attempt to recreate. In terms of performance, the simple mean imputation performs consistently across metrics, but has notable spikes in negative performance starting at 70% missing

data. It performs the best with a missing data value of 10%. The mean imputation model with outlier removed performs similarly to the simple mean imputation model, albeit with a slightly smoother trend, showing better performance than its mean model counterpart in the presence of moderate missing data (40-70%).

The best-case imputation model performs similarly to the simple mean imputation, but with a noticeable decline in R^2 scores as the missing data increases. The worst-case imputation model shows significant variability in its performance metrics, especially in its R^2 scores. This model has the worst performance overall, with high errors and low R^2 scores. Interestingly, there is a significant jump in the R^2 scores as the missing data percentage approaches 90%, which could be an anomaly or may warrant further investigation. The previous-day imputation model suffers significantly as the percentage of missing data increases, showing the worst error rates at high levels of missing data, and extremely poor R^2 score, nearing zero at 90% missing data.

The mean imputation techniques, both with and without outliers, seem to be the most stable choice for data imputation across different levels of missing data. Removing outliers doesn't drastically improve the simple mean imputation model, but it does provide a marginal benefit in terms of the R^2 score. Best-case imputation also performs reasonably well, albeit slightly less consistently. Worst-case and previous-day imputation perform poorly, especially as the level of missing data increases. Each of the models suffers significant performance deterioration when the level of missing data exceeds 50%, so more robust imputation techniques may be required for such cases. In summary of this batch of results, mean imputation with outliers removed seems to be the best choice for most metrics and levels of missing data, but if achieving the lowest MAE is the priority, then best-case imputation may be preferable due to its high performance in this category.

Table 2 shows the performance metrics of the random forest classifier on the original, un-imputed data. This is to provide a ground truth set of values to compare the impact of each of the imputation models on fault classification accuracy. Table 3 shows the performance metrics of the random forest classifier on each of the proposed imputation methods on ranges of missing data from 10%-90%. The performance metrics used are given by the following formulas:

6) Accuracy: the percentage of true positive (TP) and true negative (TN) predictions out of the total number of predictions, with false positives (FP) and false negatives (FN) included:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

7) Precision: the percentage of correctly classified positive predictions relative to the number of predictions classified as positive:

$$P = \frac{TP}{TP + FP} \tag{7}$$

8) Recall: the percentage of true positive predictions that were correctly classified as positive:

$$R = \frac{TP}{TP + FN} \tag{8}$$

9) F1-Score: the harmonic mean of the precision and recall:

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{9}$$

10) Jaccard Score: the similarity of predicted labels (PL) to true labels (TL)

$$J(PL, TL) = \frac{PL \cap TL}{PL \cup TL} \tag{10}$$

Table 2
Performance metrics for the RF classifier on the original data

RF Fault Classification Performance				
A	**P**	**R**	**F1**	**J**
99.88%	99.52%	99.83%	99.68%	99.36%

Table 3
Performance metrics for the RF classifier on the imputed data

Missing Data %									
Metric	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
	Simple Mean Imputation								
A	69.51%	68.12%	69.16%	69.51%	71.12%	72.04%	72.84%	73.19%	75.83%
P	69.59%	68.13%	69.13%	69.56%	71.23%	72.07%	72.87%	73.19%	75.83%
R	69.60%	68.16%	69.16%	69.58%	71.23%	72.10%	72.90%	73.23%	75.87%
F1	69.50%	68.11%	69.13%	69.50%	71.12%	72.03%	72.84%	73.18%	75.82%
J	53.26%	51.65%	52.84%	53.26%	55.18%	56.29%	57.28%	57.70%	61.06%
Simple Mean Imputation (Outliers Removed)									
A	68.59%	68.24%	66.40%	68.01%	70.55%	71.59%	72.17%	72.40%	75.87%
P	68.61%	68.23%	66.39%	68.05%	70.56%	71.65%	72.25%	72.43%	75.97%
R	68.61%	68.24%	66.39%	68.04%	70.57%	71.63%	72.22%	72.43%	75.92%
F1	68.59%	68.24%	66.39%	68.01%	70.55%	71.59%	72.17%	72.40%	75.86%
J	52.20%	51.79%	49.69%	51.53%	54.50%	55.75%	56.45%	56.74%	61.11%
Best-Case Imputation									
A	71.58%	74.11%	74.68%	75.26%	78.83%	80.32%	82.28%	85.16%	87.69%
P	71.86%	74.38%	74.88%	75.46%	78.94%	80.45%	82.52%	85.34%	88.00%
R	71.77%	74.29%	74.84%	75.42%	78.95%	80.45%	82.45%	85.31%	87.89%
F1	71.57%	74.10%	74.68%	75.26%	78.83%	80.32%	82.28%	85.16%	87.68%
J	55.73%	58.86%	59.59%	60.33%	65.05%	67.11%	69.89%	74.15%	78.07%
Worst-Case Imputation									
A	71.00%	72.04%	73.30%	75.49%	79.63%	82.16%	85.16%	88.15%	93.90%
P	71.08%	72.05%	73.30%	75.52%	79.63%	82.13%	85.13%	88.13%	93.88%
R	71.09%	72.09%	73.33%	75.56%	79.68%	82.15%	85.16%	88.19%	93.95%
F1	71.00%	72.03%	73.29%	75.48%	79.62%	82.14%	85.14%	88.14%	93.90%
J	55.04%	56.29%	57.84%	60.62%	66.15%	69.70%	74.13%	78.80%	88.50%
Previous-Day Imputation									
A	66.05%	64.67%	63.06%	61.57%	58.92%	55.81%	54.89%	51.78%	50.63%
P	66.23%	64.75%	63.08%	61.56%	58.98%	55.91%	55.06%	51.95%	50.80%
R	66.20%	64.76%	63.10%	61.58%	58.99%	55.91%	55.06%	51.94%	50.80%
F1	66.05%	64.67%	63.05%	61.55%	58.92%	55.81%	54.88%	51.77%	50.61%
J	49.31%	47.79%	46.04%	44.46%	41.76%	38.71%	37.82%	34.93%	33.89%

5. Conclusion

This study aimed to explore the performance of ML-based data imputation techniques in the context of CMB. Specifically, this paper sought to assess how varying methods of imputing missing data from temperature sensors would affect the overall performance of CBM strategies. Five data imputation techniques were investigated: simple mean imputation, mean imputation with outliers removed, best-case imputation, worst-case imputation, and previous-day imputation. Each was evaluated based on its ability to reconstruct missing data as well as its impact on machine fault diagnosis using a RF classifier.

The results showed a consistent trend: as the percentage of missing data increased, the performance of all imputation models generally decreased in their capability to accurately reconstruct the original dataset. Among the methods examined, mean imputation with outliers removed emerged as the most stable, followed by simple mean imputation and best-case imputation. On the contrary, worst-case and previous-day imputations displayed deteriorated performance, especially when the threshold of missing data exceeded 50%.

Interestingly, when it comes to fault classification accuracy, the worst-case imputation model outperformed other methods, particularly as higher missing data levels. This suggests that while worst-case imputation may not be suitable for data reconstruction, it might have specialized utility in classification tasks. These contrasting findings underline the complexity of choosing an optimal imputation technique, as the choice may depend on the specific metrics or operational outcomes one aims to optimize.

The findings of this study have critical implications for CBM in industrial applications. Understanding the strengths and limitations of each imputation model can guide maintenance professionals in selecting the most appropriate method for their specific needs, thereby enhancing the reliability and effectiveness of CBM strategies. However, as each of the models showed significant performance deterioration when the level of missing data exceeded 50%, future research should focus on developing more robust imputation techniques that can handle large data gaps. Moreover, the unexpected trends in classification performance at higher missing data levels warrant further investigation to understand the underlying mechanisms. Similarly, the significant jump in R^2 scores for worst-case imputation at 90% missing data could be an anomaly or a phenomenon requiring deeper study.

In summary, this research fills a gap in the existing literature by offering a comprehensive evaluation of ML-based data imputation techniques in the context of CBM. The study provides actionable insights for industry professionals aiming to bolster the robustness of their predictive maintenance strategies and introduces avenues for future research in this critical

area of industrial engineering. While the current models offer a starting point, the quest for the most effective data imputation methods in CBM remains an open challenge, underlining the need for continued investigation and innovation in this field.

References

Accorsi, R., Manzini, R., Pascarella, P., Patella, M., & Sassi, S. (2017). Data Mining and machine learning for condition-based maintenance. *Procedia Manufacturing*, *11*, 1153–1161. https://doi.org/10.1016/j.promfg.2017.07.239

Adriana Mercioni, M., & Holban, S. (2022). Prediction of machine temperature system failure using a novel activation function. 2022 International Symposium on Electronics and Telecommunications (ISETC). https://doi.org/10.1109/isetc56213.2022.10010046

Akouemo, H. N., & Povinelli, R. J. (2017). Data improving in time series using ARX and Ann Models. *IEEE Transactions on Power Systems*, *32*(5), 3352–3359. https://doi.org/10.1109/tpwrs.2017.2656939

Alabadla, M., Sidi, F., Ishak, I., Ibrahim, H., Affendey, L. S., Che Ani, Z., Jabar, M. A., Bukar, U. A., Devaraj, N. K., Muda, A. S., Tharek, A., Omar, N., & Jaya, M. I. (2022). Systematic review of using machine learning in imputing missing values. *IEEE Access, 10*, 44483–44502. https://doi.org/10.1109/access.2022.3160841

Alwan, W., Ngadiman, N. H. A., & Hassan, A. (2022). Ensemble classifier with missing data in control chart patterns. *Proceedings of the International Conference on Industrial Engineering and Operations Management*. https://doi.org/10.46254/au01.20220420

Appoh, F., & Yunusa-Kaltungo, A. (2021). Risk-informed support vector machine regression model for component replacement—a case study of Railway Flange Lubricator. *IEEE Access*, 9, 85418–85430. https://doi.org/10.1109/access.2021.3088586

Barnes, S. A., Larsen, M. D., Schroeder, D., Hanson, A., & Decker, P. A. (2010). Missing data assumptions and methods in a smoking cessation study. *Addiction*, 105(3), 431–437. https://doi.org/10.1111/j.1360-0443.2009.02809.x

Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine learning in predictive maintenance towards Sustainable Smart Manufacturing in industry 4.0. *Sustainability*, *12*(19), 8211. https://doi.org/10.3390/su12198211

Du, J., Hu, M., & Zhang, W. (2020). Missing data problem in the monitoring system: A Review. *IEEE Sensors Journal*, *20*(23), 13984–13998. https://doi.org/10.1109/jsen.2020.3009265

Emir, Ş. (2023). An investigation of anomaly detection methods in machine learning for high dimensional datasets. *Global Studies on Management Information Systems*, 227–254. https://doi.org/10.26650/b/ss28et06.2023.006.10

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hartini, E. (2017). Implementation of missing values handling method for evaluating the System/Component Maintenance Historical Data. *Journal of Nuclear Reactor Technology*, *19*(1), 11. https://doi.org/10.17146/tdm.2017.19.1.3159

Hey, J., Malloy, A. C., Martinez-Botas, R., & Lamperth, M. (2016). Online monitoring of electromagnetic losses in an electric motor indirectly through temperature measurement. *IEEE Transactions on Energy Conversion*, *31*(4), 1347–1355. https://doi.org/10.1109/tec.2016.2562029

Jha, S., Cui, S., Xu, T., Enos, J., Showerman, M., Dalton, M., Kalbarczyk, Z. T., Kramer, W. T., & Iyer, R. K. (2019). Live Forensics for Distributed Storage Systems. *arXiv*. https://arxiv.org/abs/1907.10203

Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials – A practical guide with flowcharts. *BMC Medical Research Methodology*, *17*(1). https://doi.org/10.1186/s12874-017-0442-1

Lavin, A., & Ahmad, S. (2015). Evaluating real-time anomaly detection algorithms - the Numenta Anomaly Benchmark. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. https://doi.org/10.1109/icmla.2015.141

Li, Z., & He, Q. (2015). Prediction of railcar remaining useful life by multiple data source fusion. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2226–2235. https://doi.org/10.1109/tits.2015.2400424

Loukopoulos, P., Sampath, S., Pilidis, P., Zolkiewski, G., Bennett, I., Duan, F., & Mba, D. (2016). Dealing with missing data for prognostic purposes. *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*. https://doi.org/10.1109/phm.2016.7819934

Martins, A. B., Fonseca, I., Farinha, J. T., Reis, J., & Marques Cardoso, A. J. (2022). Prediction maintenance based on vibration analysis and deep learning – A case study of a drying press supported on a hidden Markov model. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4194601

McKinney, W. (2010). Data Structures for Statistical Computing in python. *Proceedings of the Python in Science Conference*. https://doi.org/10.25080/majora-92bf1922-00a

Merkt, O. (2019). On the use of predictive models for improving the quality of Industrial Maintenance: An analytical literature review of Maintenance Strategies. *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*. https://doi.org/10.15439/2019f101

Osman, M. S., Abu-Mahfouz, A. M., & Page, P. R. (2018). A survey on data imputation techniques: Water distribution system as a use case. *IEEE Access*, *6*, 63279–63291. https://doi.org/10.1109/access.2018.2877269

Pedersen, A., Mikkelsen, E., Cronin-Fenton, D., Kristensen, N., Pham, T. M., Pedersen, L., & Petersen, I. (2017). Missing data and multiple imputation in clinical epidemiological research. *Clinical Epidemiology*, *9*, 157–166. https://doi.org/10.2147/clep.s129785

Rafsunjani, S., Safa, R. S., Imran, A. A., Rahim, S., & Nandi, D. (2019). An empirical comparison of missing value imputation techniques on APS Failure Prediction. *International Journal of Information Technology and Computer Science*, *11*(2), 21–29. https://doi.org/10.5815/ijitcs.2019.02.03

Schmidt, F. (2020). Anomaly Detection in Cloud Computing Environments. https://doi.org/10.14279/depositonce-10393

Song, C., Zheng, Z., & Liu, K. (2022). Building local models for flexible degradation modeling and Prognostics. *IEEE Transactions on Automation Science and Engineering*, *19*(4), 3483–3495. https://doi.org/10.1109/tase.2021.3124144

Song, I., Yang, Y., Im, J., Tong, T., Ceylan, H., & Cho, I. H. (2020). Impacts of fractional hot-deck imputation on learning and prediction of Engineering Data. *IEEE Transactions on Knowledge and Data Engineering*, *32*(12), 2363–2373. https://doi.org/10.1109/tkde.2019.2922638

Srimedha, B. C., Naveen Raj, R., & Mayya, V. (2022). A comprehensive machine learning based pipeline for an accurate early prediction of sepsis in ICU. *IEEE Access*, *10*, 105120–105132. https://doi.org/10.1109/access.2022.3210575

The Pandas Development Team. (2020). *pandas-dev/pandas: Pandas* (Version 2.1.1) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3509134

Wang, M., Yang, C., Zhao, F., Min, F., & Wang, X. (2023). Cost-sensitive active learning for incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(1), 405–416. https://doi.org/10.1109/tsmc.2022.3182122

Zhang, Z.-W., Tian, H.-P., Yan, L.-Z., Martin, A., & Zhou, K. (2022). Learning a credal classifier with optimized and adaptive multiestimation for missing data imputation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(7), 4092–4104. https://doi.org/10.1109/tsmc.2021.3090210