# Abstract

Completeness of data is vital for decision making and forecasting of Building Management System (BMS) datasets. Missing data can result in biased decision making down the line. This study creates a guideline for imputing the gaps in BMS datasets by comparing four methods. The four methods are from various backgrounds and are tested on a real BMS and KNMI dataset. The focus of this paper is not to impute every cell as accurate as possible but to impute trends back into the missing data. The performance is evaluated using variance error and statistics such as Kurtosis and Skewness.

# Introduction

Missing data is a common occurrence in time-series data, for this specific case causes include faulty sensors or errors in data storage. Missing data can cause downstream applications to malfunction and can thus have serious consequences. Missing data in Building Management Systems (BMS) can cause underperforming building services e.g., lower comfort of living or higher power usage, or in worst-case scenarios building breakdown as system control decisions are based on the collected data.

Imputation methods evaluated in this paper are selected from previous research that has been done into the imputation of time series data. The methods that are selected for evaluation are: Last Observation Carried Forward (LOCF), K-Nearest Neighbour algorithm, Recurrent Neural Network (RNN) and Hot Deck (HD).

Hot Deck has been outperformed by machine learning in the past as seen in (Sree Dhevi, 2014) [1] but it is applicable due to the number of similar units available for study. The time-series imputation performance of different types of RNN’s has been studied before in Che et al. (2018) [2]. The study concluded that when a Gated Recurrent Units (GRU) architecture is properly set up *“it pulled significantly ahead of non-deep learning methods”* [2].

Pazhoohesh et al. (2019) [3] found that for datasets where 10 to 30 %of the data is missing KNN regression does great compared with eight other methods. Poloczek et al. 2014 [4] analysed the use of KNN regression and FFIL and found that both did well for the study, but that KNN regression dominated other methods.

There are limited studies to clarify how to deal with missing data in BMS datasets. Previous research has focused on lighting and occupancy [3] data or created a generic framework for imputing data from multiple sensors [5]. In the case of (Zhang,2020) it is advised that a more generic plug-n-play framework is to be further studied. This study will not build on the framework created by (Zhang,2020) but tries to give a guideline as to use the methods evaluated in this paper. The research focused on imputing trends rather than accurately imputing data in a single moment in time.

This paper aims to evaluate and compare the imputation performance of the following methods: KNN algorithm, LOCF, RNN and Hot Deck. The imputation performance has been evaluated by making use of various criteria to facilitate the choice of the most suitable method for each scenario.

The method section will contain a description of the datasets, description of the pipeline, imputation methods and the criteria used for evaluation. The result section will present the imputation results and a recommended action for each data classification and gap size.

# Methodology

## Dataset description

BMS datasets store sensor data such as fluid temperature, power usage, flow rate, operational mode, solar radiation, and outdoor temperature. methods two datasets have been used, twenty-five weather stations from the Royal Netherlands Meteorological Institute (KNMI) and BMS data of hundred-twenty residential Net-Zero energy houses. The NZEB BMS time series dataset contains data from 2019 and is supposed to have five-minute interval data measurements (105096 rows). The KNMI dataset contains data from 2018 to 2020 and is measured at hourly intervals (17545 rows). The only change made to the datasets was converting the timestamps to Python Date Time objects.

## Columns selected for imputation

To get a general impression of imputation performance on other sensors and efficiency of research seven columns are selected from the two datasets to evaluate the imputation performance. The selected features from the BMS dataset are power usage (power), CO2 level measurements (CO2), heat pump flow temperature (flow\_temp) and operational mode (op\_mode). The features selected from the KNMI datasets are solar radiation (global radiation), temperature (temperature) and relative atmospheric humidity (Relative atmospheric humidity).   
The columns were selected for the classification of data and the KNMI columns were also selected for the strong correlation.

**Table 1.  
 Title:** Columns selected for imputation **Description:** Columns with the dataset of origin, device, unit of measurement and classification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Dataset** | **Device** | **Unit of measurement** | **Classification** |
| Temperature | KNMI | - | C (in 0.1c) | Interval |
| Global Radiation | KNMI | - | j per cm2 | Ratio |
| Humidity | KNMI | - | % | Ratio |
| Flow\_temp | BMS | Alklima Heat Pump | C | Interval |
| op\_mode | BMS | Alklima Heat Pump | 0-6 modes | Nominal |
| Power | BMS | Smartmeter | W | Ratio |
| C02 | BMS | C02 Sensor | PPM | Ratio |

## Pipeline

To evaluate the performance of imputation methods under the same reproducible conditions a pipeline has been developed. The pipeline performed the following tasks: loading the data, creating gaps, imputing the artificial gaps, calculating imputation performance, and storing the evaluation results. The pipeline code and trained models can be found in the appendix.

### Gap creation

To evaluate the performance of each imputation method artificial gaps are created in both datasets. The gaps come in different sizes to evaluate the performance of each imputation method the amount of sequential missing data. Gaps are created along the rules stated in the table below and are generated using a set random seed. The set random seed is also used to determine gap location and the size of the gap. The gap sizes and locations are the same for every feature and method tested.

**Table 2.**

**Title:** BMS artificial gap rules  
**Description:** BMS gap sizes with minimum size, maximum size, and percentage of total missing data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Min\_size** | **Max\_size** | **% Of data** |
| 1 | 5 min | 60 min | 15 |
| 2 | 1 hour | 6 hours | 4 |
| 3 | 6 hours | 24 hours | 1.5 |
| 4 | 24 hours | 72 hours | 0.5 |
| 5 | 72 hours | 168 hours | 0.01 |

**Table 3.**

**Title:** KNMI artificial gap rules **Description:** KNMI gap sizes with minimum size, maximum size, and percentage of total missing data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Min\_size** | **Max\_size** | **% Of data** |
| 1 | 1 hour | 6 hours | 15 |
| 2 | 6 hours | 24 hours | 5 |
| 3 | 24 hours | 72 hours | 1.5 |
| 4 | 72 hours | 168 hours | 0.005 |

## Imputation methods

Four imputation methods are compared in this paper: Hot Deck, Recurrent Neural Network (RNN), Last Observation Carried Forward (LOCF) and K-Nearest Neighbour algorithm (KNN). The methods are selected from previous literature and aim to have a wide scope of imputation approach to facilitate the characterizations, advantages, and disadvantages of each method.

### KNN algorithm

KNN algorithm is a nonparametric imputation method that works by taking the average of a gap’s K-number of neighbours. Treating every neighbouring value equally KNN would make it more vulnerable to outliers. To mitigate this KNN is set up to weigh the nearer neighbours of a gap heavier than further away values.

In Pazhoohesh et al. (2019) KNN achieved the best result according to the research when selecting the K-number in proportion to the percentage of data missing. The K-values that were tested in this paper [3] are 1, 2, 4, 6, 8 and 10, with 1 or 2 being best at 10% and 4 the best at 30% missing. For this paper, the K-value has been redetermined because of the difference in the data used.

The K-values tested are: 1,5,10,15,20,100. The K-value selection was done by evaluating the results gotten from imputation using the Variance Error. From the results of the evaluation, it can be concluded that K=20 is the best for this research.

### Last Observation Carried Forward

Last Observation Carried Forward works by filling in the gap with the last valid before the gap observation forwards. LOCF can introduce substantial bias in datasets that do have high volatility in values [6]. Columns such as power usage will most likely suffer the most from this due to the unpredictability in data which is expected to worsen with larger gaps in the data. However, LOCF is still in common use to this day and has been compared before in time-series imputation performance [3-5].

### Hot deck

#### Introduction to Hot Deck

Hot-deck imputation is a method for handling missing data in which each missing value from a recipient is replaced with an observed value from a similar unit (the donor). This method applies perfectly to this study since there are multiple units (different houses or different weather stations’ data).

This method is used extensively in practice, but the theory behind the Hot Deck is not as well developed as that of other imputation methods, leaving researchers and analysts with limited guidance on how to apply the method, the main challenge being the donor selection.

In some versions, the donor is selected randomly from a set of potential donors, which is called the donor pool. In other, more deterministic, versions a single donor is identified, and values are imputed from that case, usually the “nearest neighbour” based on a dataset-dependent metric (i.e.: the mean when imputing temperature time series).

#### Implementing the donor selection

##### In theory

In the case of this research, the donor selection was based on pattern recognition.

It works by taking an extract containing data before and after a series of missing values (a gap) found in the recipient.  
  
To find the best matching segment of data from a donor, the recipient’s extract would then be compared to similarly sized extracts from the same time period in a donor.

Using the difference in the mean of the donor’s extracts and recipient’s extract, the values from the donor’s extracts can be shifted towards those of the recipient — except when imputing classification data —.  
  
The sum of the absolute difference between the extracts can now be used to sort the comparisons: the smaller the sum, the better the pattern matches.  
  
The operation can then be repeated throughout each donor of the donor pool, for each gap, to find the best possible match before finally importing data into the recipient.

#### In application

This donor selection method has been applied in two versions for this paper.

Whereas the first iteration had a focus on precision, using interpolation to have the most accurate value between two data measurements, for example. The second iteration focused on improving processing time, by vectorizing the search algorithm.

But the processing time improvements had a negligible cost in precision. Which was even more diminished by the ability to compare the recipient’s extract to superior amounts of data from each donor for every gap (equivalent to a month plus the gap size).

### Recurrent Neural Network

*TBA*

**Table 4.  
Title:** List of methods included in this paper **Description:** Methods used with an abbreviation, category a short description and Python library of origin.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Abbreviation** | **Category** | **Description** | **Library used** |
| Forward filling | LOCF | Simple | Use the last cell before the gap to fill a gap | Pandas.DataFrame.fillna |
| KNN regression | KNN | Simple | Take the weighted average K-number of nearest neighbours. | Sklearn.impute.KNNImputer |
| GRU RNN | RNN | Neural Network |  |  |
| Hot deck | HD | Statistical | Take data from a different unit with a similar trend. | **None** |

## Imputation evaluation criteria

The research aims to create a selection of the most suitable imputation methods for certain scenarios, it is crucial to use multiple criteria to evaluate the performance of each method. The selected criteria to evaluate each method with are Variance Error (VE), Percent Bias (PB) and the impact on both Kurtosis and Skewness when compared to original data.

VE is selected to give insight into the imputation method’s ability to impute a trend back into the missing data. The VE is calculated by taking the variance of the original data in the gap and the imputed data and then calculating the difference. If there are multiple gaps in a dataset the average of the difference in variance is taken as the VE. The library used to calculate the variance is pandas.

PB is used to see if an imputation method tends to over-or underestimate data compared to the original data. Bias in imputation can worsen forecasting when the bias has a large presence or is high enough. Evaluating an imputation method based on its PB can help give insight into the imputed data’s performance when using it to forecast.

To give insight into the effect of imputation on the missing data compared to the original data Skewness and Kurtosis are used. These statistical values can put the change in distribution and spikiness from original to imputed data into context. Because predicting trends back into missing data is a focal point of this paper these statistical values are used to strengthen the evaluation along with the variance.   
The Kurtosis and Skewness are calculated using scipy.stats. kurtosis/skew.

# Results

To make this research applicable to BMS datasets outside this paper the focus will be on data classifications of columns instead of the columns themselves. From the results the following conclusions can be made:

* Conclusion point #1:

|  |  |  |  |
| --- | --- | --- | --- |
| BMS | **Interval** | **Ratio** | **Nominal** |
| **Gap Type 1** |  |  |  |
| **Gap Type 2** |  |  |  |
| **Gap Type 3** |  |  |  |
| **Gap Type 4** |  |  |  |
| **Gap Type 5** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| KNMI | **Interval** | **Ratio** | **Nominal** |
| **Gap Type 1** |  |  |  |
| **Gap Type 2** |  |  |  |
| **Gap Type 3** |  |  |  |
| **Gap Type 4** |  |  |  |

# Future work

In future work, the focus of research should be less on evaluating imputation with metrics based on the error but on its impact on forecasting using imputed data. The effect on forecasting performance ought to be evaluated as it can provide a more complete view of imputation performance.

This dataset was limited to only numerical data so further testing on the performance on non-numerical ought to be considered too for future work.

Whilst this paper contains a guideline of what imputation methods to use in what scenario of data classification and gap size it could be improved in the form of an automatic framework. This would provide an easier solution to the problem of missing data in BMS time series data.

# References

1. Sree Dhevi, A. T. (2014). Imputing missing values using Inverse Distance Weighted Interpolation for time series data. 2014 Sixth International Conference on Advanced Computing (ICoAC). <https://doi.org/10.1109/icoac.2014.7229721>
2. Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. Scientific Reports, 8(1). <https://doi.org/10.1038/s41598-018-24271-9>
3. Pazhoohesh, M., Pourmirza, Z., & Walker, S. (2019). A Comparison of Methods for Missing Data Treatment in Building Sensor Data. 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE). <https://doi.org/10.1109/sege.2019.8859963>
4. Poloczek, J., Treiber, N. A., & Kramer, O. (2014). KNN Regression as Geo-Imputation Method for Spatio-Temporal Wind Data. Advances in Intelligent Systems and Computing, 185–193. <https://doi.org/10.1007/978-3-319-07995-0_19>
5. Zhang, L. (2020). A Pattern-Recognition-Based Ensemble Data Imputation Framework for Sensors from Building Energy Systems. Sensors, 20(20), 5947. <https://doi.org/10.3390/s20205947>
6. Little, R., & Yau, L. (1996). Intent-to-Treat Analysis for Longitudinal Studies with Drop-Outs. Biometrics, 52(4), 1324–1333.

<https://doi.org/10.2307/2532847>