

## Abstract

Completeness of data is vital for the decision making and forecasting of Building Management System (BMS) as 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: K-Nearest Neighbours algorithm (KNN), Recurrent Neural Network (RNN), Hot Deck (HD) and Last Observation Carried Forward (LOCF). The guideline contains the best method per gap size and scales of measurement. The four selected methods are from various backgrounds and are tested on a real BMS and meteorological dataset. The focus of this paper is not to impute every cell as accurately as possible but to impute trends back into the missing data. The performance is characterised by a set of criteria in order to allow the user to choose the imputation method best suited for its needs. The criteria are: Variance Error (VE) and Root Mean Squared Error (RMSE). VE has been given more weight as its ability to evaluate the imputed trend is better than RMSE. From preliminary results, it was concluded that the best K-values for KNN are 5 for the smallest gap and 100 for the larger gaps. Using a genetic algorithm the best RNN architecture for the purpose of this paper was determined to be Gated Recurrent Units (GRU). The comparison was performed using a different training dataset than the imputation dataset. The results show no consistent link between the difference in Kurtosis or Skewness and imputation performance. The results of the experiment concluded that RNN is best for interval data and HD is best for both nominal and ratio data. There was no single method that was best for all gap sizes as it was dependent on the data to be imputed.

**Keywords:** Building Management System time series data, Imputation, KNN, RNN, Hot Deck, trend

## 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 (KNN), Recurrent Neural Network (RNN) and Hot Deck (HD).

HD 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, the KNN algorithm does great compared to eight other methods. Poloczek et al. 2014 [4] analysed the use of KNN regression and LOCF and found that both did well for the study, but that KNN regression outperformed 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 does not build on the framework created by (Zhang,2020) but tries to give a guideline on when to use what method. 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. Aside from the most suited scenario, the imputation method's ability to impute trends is also evaluated.

The method section contains a description of the datasets, description of the pipeline, imputation methods and the criteria used for evaluation. The result section presents the imputation results and a recommended action for each data measurement scale 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. Two datasets have been used: BMS data of one hundred twenty residential Net-Zero energy houses and twenty five weather stations from the Royal Netherlands Meteorological Institute (KNMI) [8]. 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, 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 BMS features were selected for their data measurement scale. The KNMI columns were selected for the strong correlation between the features.

**Table 1.**

**Title:** Columns selected for imputation

**Description:** Columns with the dataset of origin, device, unit of measurement and measurement scale.

Column name	Dataset	Device	Unit of measurement	Measurement scale
Temperature	KNMI	-	C (in 0.1c)	Interval
Global Radiation	KNMI	-	j per cm <sup>2</sup>	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
CO2	BMS	CO2 Sensor	PPM	Ratio

## Pipeline

A pipeline has been developed to evaluate the performance of imputation methods under the same reproducible conditions. 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 references.

## 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 on different amounts of missing sequential data. Gaps are created making use of the rules stated in the table below and are generated using a set random seed. The set random seed is also used in order 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 have been selected from previous literature and aim to have a wide scope of imputation approaches to facilitate each method's characterizations, advantages and disadvantages.

### 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, simple 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.

The K-values tested are: 1,5,10,15,20,100. The K-value selection has been done by evaluating the results gotten from imputation using the Variance Error. From the results of the evaluation, it can be concluded that K=5 is best for the gap size 1 and K=100 for gap sizes 2 to 5.

### 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 are most likely to suffer the most from this due to the unpredictability in data which is expected to worsen with larger gaps in the data. Nevertheless, LOCF is still in common use nowadays and has been compared before in time series imputation performance [3-5].

## 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).

HD is a well-known method, but the theory behind the Hot Deck is not as well developed as the theory of other imputation methods, leaving researchers with limited guidance on how to apply it. The main challenge is selecting donors.

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).

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 categorical data. The sum of the absolute difference between the extracts can now be used to sort the comparisons: the smaller the sum is, 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.

## Recurrent Neural Network

Recurrent Neural Networks have been proven to perform well when working with time series data [7] and in [2] it pulled significantly ahead of non-deep learning methods. RNN's benefits from having an internal memory, unlike other NNs, this helps them to preserve context which is useful with the imputation of BMS time series data. The internal memory of the RNN architecture is useful for the purpose of imputing time series data as the missing values are highly dependent on the trend before and after a gap.

Two different architectures of RNN's were compared on performance in time series imputation: Long-Short term Memory (LSTM) and Gated Recurrent Units (GRU). Both architectures use long-short term memory, but the key difference is that LSTM uses three gates: forget, input and output, whilst GRU uses two: update and reset. Another difference between the two is that GRU exposes the entire memory including hidden layers whilst LSTM keeps them hidden.

Both architectures were compared using a genetic algorithm, random configurations are generated by changing the GRU or LSTM. Architectural parts that were randomized are the number (1 – 5) and size (2 – 100) of hidden layers, input sequence size (2 – 12) and the loss-function (MSE or Huber).

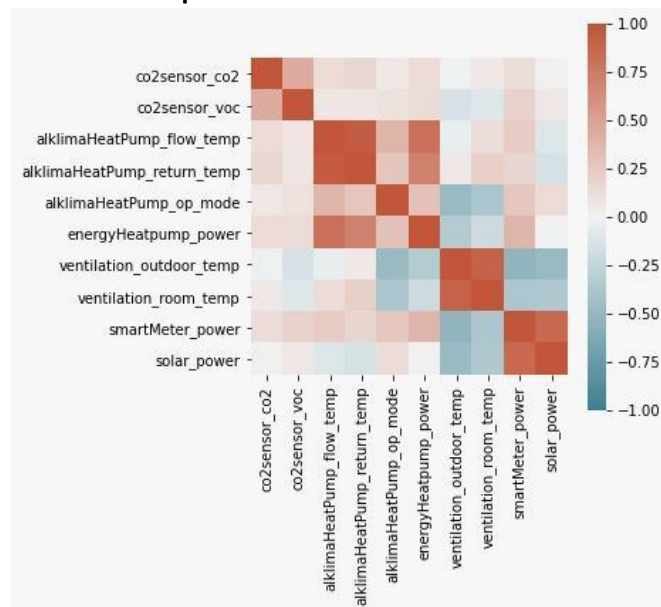
The best architecture configuration found during testing was a GRU RNN configured with 1 hidden layer of size 95, input sequence length 12 and the using the MSE loss function. The final GRU was configured as listed before in addition a fully connected layer was added. This was done to transform the GRU layer output into prediction. The GRU-based RNN was trained for every column that was to be imputed.

RNN results are drastically improved when using multiple correlated columns as input. To select the most correlated columns a correlation matrix (Figure 1) is used to display both positive and negative correlations. From Figure 1 the best correlators were chosen to train each model with.

**Figure 1.**

**Title:** Correlation matrix BMS data

**Description:** Predict correlation matrix BMS data



The implementation of the current GRU-based model has two major limitations. First, it only imputes one value at a time based on the X-number of preceding values. This means that with a sufficiently large gap it uses its own values to impute further. Using previously imputed values can result in biases in the imputation since one imputation error impacts all following imputations. Another limitation is that the current GRU RNN version trained, using Mean Squared Error (MSE), may not line up with the goal of imputing trends back into missing data.



**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
Last Observation Carried Forward	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	RNN considers past values to impute missing data.	torch.nn.GRU
Hot deck	HD	Statistical	Take data from a different unit with a similar trend.	None

### Imputation evaluation criteria

The aim of this paper is to create a selection of the most suitable imputation methods for certain scenarios with measurement scales and gap sizes. To select the best method for each scenario evaluation criteria are required for this paper the selected criteria are Variance Error (VE) and Root Mean Squared Error (RMSE).

We use VE to give insight into the imputation method's ability to impute trends back into the missing data since that is one of the focal points of our research. The VE is calculated by calculating the difference in variance between the original and imputed data for each gap and then averaging it out if multiple gaps are present. To get the difference in variance in a gap, the pandas method "pandas.var" has been used.

In previous literature [3-7-11] RMSE has been used to evaluate the performance of imputation on time series data. RMSE is calculated to give a comparison point for imputations done in this paper compared to results in previous research. The RMSE was calculated by taking the square root of the Mean Squared Error.

In this study normality testing of the datasets is done to measure the impact of a change in the distribution on imputation. Kurtosis and Skewness are used to test the normality of data because according to [11] it is significant to model development. According to the central limit theorem, the distribution of data can be ignored with hundreds of observations [9]. But according to [10] statistics such as Kurtosis and Skewness can be used to measure the normality of continuous data with sample sizes higher than 50.

## Results and discussion

With the developed pipeline and datasets as described in the methodology, an experiment ran with the settings of Tables 2 and 3 to determine the best imputation method per gap size and measurement scale.

The imputation target for the experiment was unit 099 for BMS data and de Bilt weather station as KNMI climate data. To train the models RNN used unit 054 and Rotterdam KNMI weather station. RNN and KNN both imputed `op_mode` with decimal values and should thus be ignored. Whilst `op_mode` is numerical, the performance of RNN and KNN wouldn't represent the performance on other text-based data.

The pipeline evaluated each imputation method based on the evaluation metrics listed under the evaluation criteria per gap and feature. The imputation performance is mainly evaluated based on the VE metric as mentioned in the evaluation criteria. RMSE is also calculated to evaluate imputation performance as seen in previous literature.

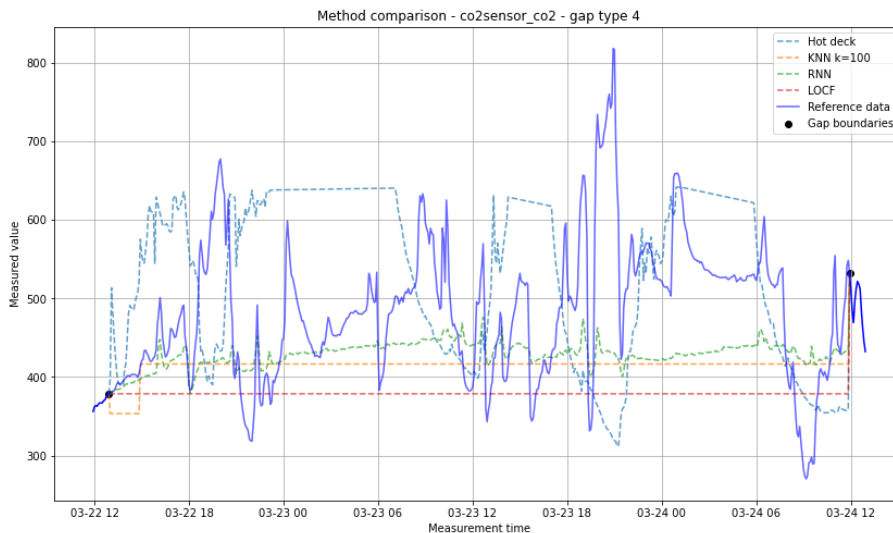
Based on the information in Table 6-10, several conclusions can be made by comparing the performance of the imputation method over various gap sizes and measurement scale:

- The RMSE results show that the imputation performance as judged by traditional metrics is poor. However, since the focus of this paper is to impute trends the comparison point should be the VE between methods instead of the RMSE.
- HD tends to score better on the KNMI datasets, this might be due to the more similar previous trends found in the datasets. In both VE and RMSE, HD tends to outperform RNN in imputing temperature data across all gaps. This performance doesn't transition to other interval data such as `flow_temp` as HD's performance in both VE and RMSE is worse than expected in this column.
- RMSE and VE do not always align when it comes to trend prediction in imputed data. CO2 sensor data gap 3 and 4 have a relatively close RMSE while their VE scores further apart. When visualising the data in Figure 1 it can be seen that HD tries to impute a trend that matches relatively well according to the VE. However, RNN has a more stable line without any big noticeable trends. In this case, RMSE seems to punish imputing trends and reward imputing a stable line of data without trends.

**Figure 2.**

**Title:** CO2 sensor trend imputation performance comparison

**Description:** VE is higher for RNN but RMSE is about an equal trend in visual

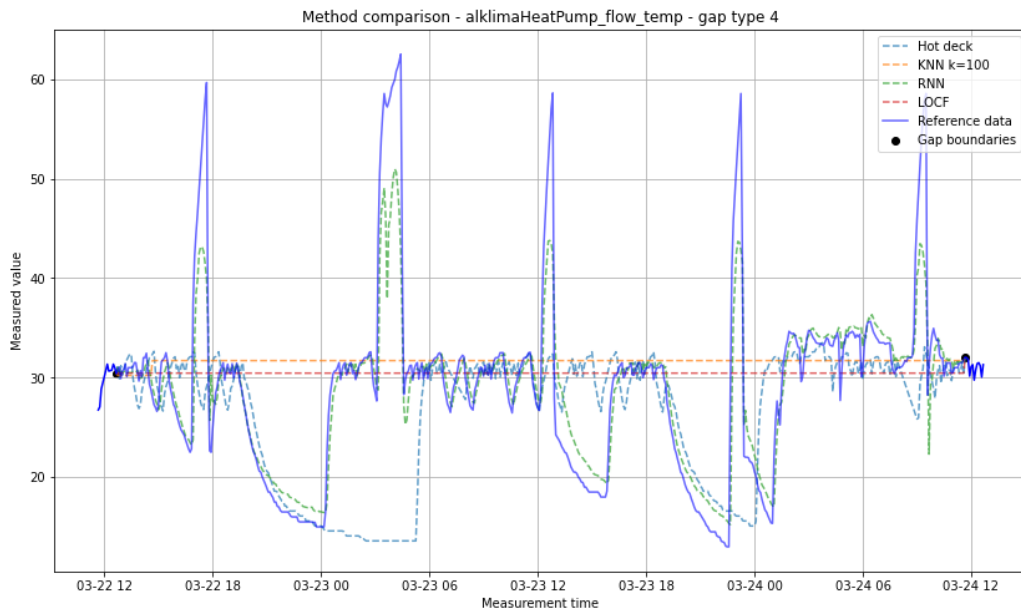


- As seen in Table 11, a high difference in Kurtosis was found between training and imputation target data. This might explain RNN's behaviour in the imputation results of the CO2 sensor data as seen in Figure 1 and Table 6-10. However, no consistent link has been found between Kurtosis and trend imputation performance. In Figure 2 the imputation of a flow\_temp gap is shown, in the figure RNN tracing a trend is observed. RNN has a comparatively better performance in flow\_temp than in power despite having a higher difference in both Kurtosis and Skewness. The normality of the distribution doesn't seem to explain the difference in RNN imputation performance from interval to ratio data.
- In the achieved results (Table 6-10) no strong link can be found between having multiple strong correlators and a good RMSE or VE score. Flow\_temp has two strong correlators in return temperature (0.94) and heat pump power usage (0.81) and gets good imputation results. Power and CO2 data seem to contrast these findings as power has two strong correlators (-0.56 and 0.86) and CO2 one decent correlator (0.44) but both get bad imputation results.

**Figure 3.**

**Title:** Trend performance RNN on interval flow\_temp data

**Description:** RNN performance in interval data



To sum up, the results as seen in Table 6-10 and discussed before there is no single imputation method for all data measurement scales. Based on the results it can be concluded that there is no imputation method that works well for all gap sizes. HD tends to score better on KNMI data across measurement scales RNN doesn't do the same. VE and RMSE don't always align and VE is in some cases the better indicator for the ability of an imputation method to follow the trend displayed in Figures 1 & 2. The difference in Kurtosis wasn't as big of a factor as initially thought.

## Conclusion

This paper proposes a guideline to impute BMS nZEB data based on gap size and different scales of measurement.. The problem with missing data in BMS is becoming a bigger problem in an era where buildings depend on data. Previous research has been done about imputing BMS time series data; this paper tries to build on that by creating a comprehensive guideline to follow for certain scenarios. To create a guideline 4 methods were chosen from previous literature: GRU RNN, Hot Deck, KNN algorithm and LOCF. During the research, imputing trends back into missing data became the focal point of this study which is why Variance Error (VE) was used instead of a more traditional metric like Root Mean Squared Error (RMSE). The guideline that resulted from this experiment is listed down below in Table 5. Performance was evaluated using both RMSE and VE but metrics concluded the same methods as best for each gap type and data measurement scale.

From the results of both VE and RMSE can be concluded that there is no single best imputation method for all gap sizes and measurement scales. The best method for a gap size is dependent on the measurement scale of the to be imputed data. No consistent crossover was found between the gap size and measurement scale as can be seen in Table 5.

**Table 5**

**Title:** Guideline for what method to use based on VE

**Description:** Guideline, method listed per Gap type and measurement scale

	Gap type 1.	Gap type 2.	Gap type 3.	Gap type 4.	Gap type 5.
Nominal	HD	HD	HD	HD	HD
Ratio	HD	HD	HD	HD	HD
Interval	RNN	RNN	RNN	RNN	RNN

When comparing RMSE scores achieved in this paper to previous research it seems the imputation performance is poor. The focal point of this paper however was to impute trends back into data and seeing the VE score and visualisations a good starting point was made.

An important thing to note for this paper is the use of HD, due to this research having large amounts of similar data units HD was uniquely applicable. With no similar data HD will not be applicable and even with fewer external data sets HD might suffer a performance hit.

## Future work

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

The data sets used for this study contain only numerical data and no ordinal data. To get a full view of the imputation performance on text-based categorical data further research is required.

The GRU RNN architecture used in the research had clear limitations based on how it was set up. To evaluate the full potential of imputation using RNN the architecture should be changed to an encoder-decoder sequential based design. This would remove the potential bias of imputation using its own imputed values.

**Note:** All the pipeline code, models and raw data are to be found on Github links in the references.

## Appendix

**Table 6-10. Title:** Results per feature and method in both VE and RMSE

Field	Method	HD	RNN	KNN	LOCF
<b>Gap type 1</b>					
Temperature	VE	60.13	63.738	92.126	92.701
	RMSE	14.07	12.31	38.19	22.352
FLOW_TEMP	VE	8.67	7.20	10.76	10.76
	RMSE	5.29	3.21	5.34	6.79
op_mode	VE	0.08	0.08	0.08	0.08
	RMSE	0.49	0.45	0.45	0.56
Global Radiation	VE	337.621	369.87	620.279	620.85
	RMSE	25.16	29.84	66.47	53.02
Humidity	VE	15.38	15.45	20.71	20.80
	RMSE	6.78	6.19	14.52	9.88
Power	VE	137217	151494	158763	158763
	RMSE	686.24	798.725	632.54	800.01
CO2	VE	372.92	393.49	420.02	420.02
	RMSE	44.41	40.34	35.29	43.03
<b>Gap type 2</b>					
Temperature	VE	264.83	290.36	604.13	609.78
	RMSE	17.61	17.19	38.72	45.14
FLOW_TEMP	VE	32.892	21.36	39.59	39.631
	RMSE	8.03	3.78	8.43	10.28
op_mode	VE	0.28	0.32	0.33	0.33
	RMSE	0.84	0.84	0.83	0.99
Global Radiation	VE	1086.88	1427.15	2902.85	2904.88
	RMSE	32.52	45.92	67.81	91.24
Humidity	VE	51.67	59.07	108.11	108.60
	RMSE	9.08	9.45	15.12	18.78
Power	VE	357663	425962	494539	504114
	RMSE	895.32	1258.1	1176.21	1181.55
CO2	VE	1393.59	1587.98	1782.42	1747.18
	RMSE	53.07	66.71	78.15	74.81
<b>Gap type 3</b>					
Temperature	VE	328.88	381.24	901.32	912.82
	RMSE	14.39	17.19	37.874	45.98
FLOW_TEMP	VE	53	32.168	65.99	65.99
	RMSE	10.35	3.87	9.26	11.58
op_mode	VE	0.62	0.544	0.63	0.63

	<b>RMSE</b>	1.20	0.93	0.96	1.34
Global Radiation	<b>VE</b>	1138.15	2380.38	4392.39	4396.39
	<b>RMSE</b>	28.90	57.54	68.1	96.89
Humidity	<b>VE</b>	69.05	89.98	176.42	177.53
	<b>RMSE</b>	7.64	10.26	14.75	18.64
Power	<b>VE</b>	579467	1.15337e+06	1.36275e+06	1.36607e+06
	<b>RMSE</b>	997.66	1516.92	1725.46	1906.12
C02	<b>VE</b>	3862.98	4608.09	5272.42	5282.86
	<b>RMSE</b>	109.88	100.2	113.50	115.28
	<b>Gap type 4</b>				
Temperature	<b>VE</b>	415.57	407.244	1220.58	1228.37
	<b>RMSE</b>	14.67	18.23	42.08	52
FLOW_TEMP	<b>VE</b>	54.81	33.72	73.699	73.701
	<b>RMSE</b>	9.59	3.76	9.37	11.75
op_mode	<b>VE</b>	0.47	0.67	0.78	0.78
	<b>RMSE</b>	1.20	0.98	0.96	1.34
Global Radiation	<b>VE</b>	735.97	1715.78	4221.93	4222.48
	<b>RMSE</b>	22.94	64.08	65.86	97.46
Humidity	<b>VE</b>	67.07	95.86	187.31	187.40
	<b>RMSE</b>	7.12	11.46	15.21	19.94
Power	<b>VE</b>	760667	1.59309e+06	1.96192e+06	1.96194e+06
	<b>RMSE</b>	843.932	1744.81	1759.17	1992.42
C02	<b>VE</b>	6316.95	8492.07	9311.84	9315.94
	<b>RMSE</b>	115.84	116.52	123.39	142.02
	<b>Gap type 5</b>				
FLOW_TEMP	<b>VE</b>	63.54	36.65	75.85	75.85
	<b>RMSE</b>	9.12	3.95	9.48	12.60
op_mode	<b>VE</b>	0.15	0.152	0.598	0.598
	<b>RMSE</b>	0.89	1.02	1.59	0.92
Power	<b>VE</b>	746953	1.81726e+06	2.079629e-6	2.079629e-6
	<b>RMSE</b>	778.901	1594.85	1628.55	2218.56
C02	<b>VE</b>	5956.65	9086.66	9902.27	9902.33
	<b>RMSE</b>	105.32	122.10	114.78	131.26



**Table 11.****Title:** Kurtosis and Skewness values of training and imputation target**Description:** kurtosis and skewness comparison between two different datasets (two houses for BMS data, two weather stations for KNMI data)

Column	Kurtosis for station 344 Rotterdam	Kurtosis for station 260 De Bilt	Kurtosis difference	Skewness for station 344 Rotterdam	Skewness for station 260 De Bilt	Skewness difference
Temperature	-0.22	-0.26	<b>0.04</b>	0.21	0.24	<b>0.03</b>
Relative atmospheric humidity	0.27	-0.01	<b>0.28</b>	-0.89	-0.85	<b>0.04</b>
Global radiation	2.23	2.32	<b>0.09</b>	1.76	1.78	<b>0.02</b>
Column	Kurtosis for house 54	Kurtosis for house 99	Kurtosis difference	Skewness for house 54	Skewness for house 99	Skewness difference
alklimaHeatPump flow_temp	3.20	2.75	<b>0.45</b>	1.67	1.42	<b>0.25</b>
alklimaHeatPump op_mode	5.78	3.58	<b>2.20</b>	2.42	1.60	<b>0.81</b>
smartMeter power	1.74	1.43	<b>0.31</b>	-1.01	-1.08	<b>0.07</b>
co2sensor co2	7.65	2.24	<b>5.41</b>	1.91	1.33	<b>0.57</b>

## 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](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>
7. Ma, J., Cheng, J. C., Jiang, F., Chen, W., Wang, M., & Zhai, C. (2020). A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data. Energy and Buildings, 216, 109941. <https://doi.org/10.1016/j.enbuild.2020.109941>
8. KNMI. (n.d.). KNMI - Uurgegevens van het weer in Nederland. KNMI - Hourly weather data of Dutch climate. <https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens>
9. Altman, D. G., & Bland, J. M. (1995). Statistics notes: The normal distribution. BMJ, 310(6975), 298. <https://doi.org/10.1136/bmj.310.6975.298>

10. Gupta, A., Mishra, P., Pandey, C., Singh, U., Sahu, C., & Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. *Annals of Cardiac Anaesthesia*, 22(1), 67. [https://doi.org/10.4103/aca.aca\\_157\\_18](https://doi.org/10.4103/aca.aca_157_18)
11. Shapi, M. K. M., Ramli, N. A., & Awalin, L. J. (2021). Energy consumption prediction by using machine learning for smart building: Case study in Malaysia. *Developments in the Built Environment*, 5, 100037. <https://doi.org/10.1016/j.dibe.2020.100037>
12. Pipeline used in research(2021)[source code & results].  
<https://github.com/thuas-imp-2021/thuas-imp-2021>