Journal of Building Engineering 44 (2021) 103243

Contents lists available at ScienceDirect

Journal of Building Engineering

journal homepage: www.elsevier.com/locate/jobe

A method and analysis of predicting building material U-value ranges through geometrical pattern clustering

S. Klemp [a], A. Abida [a,b], P. Richter [a,*]

[a] RWTH Aachen University, Research Group Theory of Hybrid Systems, Department of Computer Science, Ahornstr. 55, 52074, Aachen, Germany
[b] MeteoViva GmbH, Karl-Heinz-Beckurts-Str. 13, 52428, Jülich, Germany

ARTICLE INFO

Keywords:
Predictive clustering
Evaluation of building clustering
Unsupervised learning
Value range prediction
Building data preprocessing

ABSTRACT

For the optimization of the energy consumption in buildings, a calibrated model is of paramount importance. To calibrate the model, initial value ranges for unknown parameters must be defined which is often done through manual tuning and engineering methods. These values are often inaccurate or not available, thus set arbitrarily. Therefore, in this paper, we examine the possibility of defining thermodynamic value ranges by clustering geometrical building patterns. Two issues are analyzed by the method, building pattern clustering via machine learning and the predictive ability of geometrical clusters. The method involves testing multiple clustering algorithms on features extracted from calibrated commercial buildings. The algorithms are either executed on an untransformed or Box-Cox transformed feature space and then evaluated by their geometrical patterns and U-value ranges. For the assesment of U-value ranges, two new evaluation indices are introduced. The shared nearest neighbor algorithm turns out to be the most promising one for clustering geometrical data, reducing initial U-value ranges by 50% on average. In some applications, it might be undesirable to use the shared nearest neighbor algorithm, as data points are assigned as noise. For these cases a Box-Cox transformation of the data is necessary. Without a transformation, other algorithms were not able to determine any geometrical patterns. The method shows the possibility of determining unique U-value ranges by using geometrical data only. The application of such machine learning approaches enables saving time in determining initial value ranges and further the possibility of accelerating calibrations, as smaller value ranges are used.

1. Introduction

Approximately 40% of the European Union's energy consumption is needed by commercial and residential buildings [1]. Globally, buildings contribute for over 30% of the CO_2 emissions due to fossil driven energy production. A large proportion of this energy is provided for *HVAC* systems, which are used for heating, ventilation and air conditioning [2].

In order to reduce this amount, several methods gained popularity over the last years. One of them is the smart controlling of HVAC systems, which enables changes in real-time according to the demand. This is often used in conjunction with measuring devices, which record important data like the temperature or humidity consistently. That conjunction offers a huge potential in terms of energy reduction. One of the main challenges is, that in order to use such devices effectively, accurate impact predictions of changes in the HVAC system have to be performed before applying them to real buildings:

The data-driven and machine learning *black box* approach uses only the measured data of the building, while in contrast to this, the *white box* approach uses only the physical data of the building. The *grey box* approach is a combination of both. For the grey and white box approach, a virtual model of the building has to be defined, which is then used for control predictions. More accurate virtual models result in better predictions and therefore a minimization in energy loss. The accuracy of the model can be optimized via the calibration of several uncertain parameters, which describe thermodynamic properties of the building. In general, these parameters are inaccurate or at worst not known. To accelerate the calibration process and therefore also the accuracy of these models, the determination of relevant value ranges for these uncertain parameters is needed [3].

1.1. Related work

Altogether, we regard two related problems: the clustering of buildings and the value range definition which especially focuses on

* Corresponding author.
E-mail address: pascal.richter@rwth-aachen.de (P. Richter).

https://doi.org/10.1016/j.jobe.2021.103243
Received 25 April 2021; Received in revised form 26 August 2021; Accepted 1 September 2021
Available online 4 September 2021

List of acronyms and abbreviations

Algorithms

PCA	Principle component analysis
DBSCAN	Density based spatial clustering of applications with noise
SNN	Shared nearest neighbor algorithm
GMS	Mean shift algorithm with gaussian kernel
FMS	Mean shift algorithm with flat kernel

Parameter ratios

FTZV	Floor area to zone volume ratio
WWR	Window to wall ratio
RTFR	Roof to floor ratio
VRTFR	Virtual roof to floor ratio
GRTFR	Glass roof to floor ratio
IWTFR	Inner wall to floor ratio
SF	Shape factor

Approach based metrics

DB	Davies-Bouldin measure
CH	Calinski-Harabasz measure
SSC	Silhouette measure

Application based metrics

MRCS	Mean range cluster score
RPS	Range percentage score
MSSE	Mean sum of squared error

ranges for the initialization of the calibration.

1.1.1. Building clustering

So far building clustering was performed in a huge variety with different purposes. Gangolells et al. [4] presented a method to identify a set of representative buildings of an entire stock by clustering. They applied the *k*-Means algorithm to an energy performance certificate database. Seven representative office blocks in industrial buildings and nine representative offices in residential buildings have been identified. Nikolaou et al. [5] propose a method for clustering heating and cooling energy demand and the PMV index based on modelled buildings. They use hierarchical clustering, *k*-Means, Gaussian mixtures, Fuzzy C-Means and Neural SOM clustering as clustering algorithms. The validation is performed with the Silhouette, Davies-Bouldin, and Dunn index. An analysis was performed to determine properties that define a cluster of buildings. Filogamo et al. [6] propose a method to identify representative buildings of large building stocks in terms of energy consumption. They use static parameters like geometrical, thermal-physical characteristics based on the construction period, heating/cooling system type and the climate zone of the building in order to identify the sample buildings. They identified samples as 'virtual' buildings, which means, they don't exist in the real world but are just the averages of each class. Satre-Meloy et al. [7] used a cluster method to capture temporal variation patterns in electricity consumption. After that, they used classification models to predict the cluster membership. They used random forests and logistic regression for the classification. The results show, that one of the most influences on system-wide peaks are people that have earlier cooking and meal times. Hecht et al. [8] classified building footprints based on their building type. They pre-processed the data using a PCA reduction to highly correlated features and classified the data afterwards. 16 different machine learning classifiers were used, achieving accuracies of 76–95%, depending on the database type of the building footprints. Tardioli et al. [9] proposed a method for predictive classification. They classified buildings based on a variety of properties, such as geo-references, urban infrastructure, territory information, building topology, and geometric information. After an initial classification, the data was normalized by several different methods as scaling, centring, z-score, PCA, and Box-Cox. Then, a clustering based on *k*-Means, Agglomerative, Partitioning around metroids, and Divisive clustering was performed. Lastly, the results were evaluated on a normalized index composed of seven clustering validation indices.

1.1.2. Value range definition

In the literature multiple ways were proposed to obtain ranges for the initialization of the calibration. Zuhaib et al. [10] collected data of an education building from building audits, construction specifications, technical surveys, and local weather stations. The minimum and maximum values of the composed data set were then used to define the initial value range for the energy building model. Similarly, Chong et al. [11] defined the initial parameter ranges based on the minimum and maximum of measured data, as-built drawings, and specifications. For the cases where just little prior knowledge on the buildings in scope was available, the ranges had to be defined in other ways. Kristensen et al. [12] and Chong and Menberg [13] defined expected values for uncertain parameters based on prior beliefs. These guesses were then used as the mean of a data distribution which defined the ranges. Other studies [3, 14,15] use reference buildings, building norms or previous research articles to define the uncertain value ranges.

1.2. Our contribution

There exists a research gap for clustering of buildings. So far, it was rarely performed on large, high resolution data sets of real buildings. Accordingly, clustering performances and distributions were rarely, if at all, compared on a high scale of real building data, especially already calibrated one. The possibility of automatically predicting value ranges based historical data and machine learning on large data sets also lack analysis. The objective of this paper is a case study on a large data set of existing real world buildings to determine and learn patterns in the geometrical properties of clusters. Moreover, the possibility of predicting initial value ranges for calibration is analyzed using the geometrical building clusters. Although semi-supervised learning sounds like a powerful approach, we had the task to identify a strategy that aims to produce accurate result without referring to the thermodynamic know-how or engineering experience. Our goal was to investigate the unsupervised solution to avoid the partial selection of labels in the semi supervised learning approach.

1.3. Outline

A structural overview of the proceeding work is given in the following. First, in Section 2, the theoretical background for the proposed method is given. Therefore, the pre-processing methods, AI and clustering algorithms, and validation indices are explained. In Section 3, the data and the different input features are presented. The method is explained that was used to cluster the building data and analyze the possibility of thermodynamical value range prediction. Afterwards, in Section 4, the results are presented. The impact of different pre-processing methods and input data are compared and an evaluation of the validation indices is performed. Moreover, the different cluster outcomes are compared by their thermodynamical value ranges and the most promising cluster is examined further. Finally, in Section 5 a conclusion is presented and approaches for future work are given.

2. Theoretical background

In this section, the theoretical foundation required for the applied method is built. Preprocessing methods, clustering algorithms, and validation indices used are described.

Let

$$X = \{x_i \mid i \in [0, k-1]\} \tag{1}$$

be the set of data points to be clustered, let

$$C = \{c_0, c_1, \ldots, c_{n-1}\} \tag{2}$$

be the set of clusters and $\overline{c_j}$ be the centroid of cluster c_j.

2.1. U-value definition

In practice when a new building should be optimized, only the geometrical data are available. In order to make predictions and work with the building, the thermodynamical properties are needed. This can be, for example, the U-Value of a window, which describes the thermal transmittance, i.e. how much heat is transferred through the window. The U-Value is a property of a material, its units are (W/m^2 K). This implies, if a divider material had a U-Value of 1 W/m^2 K, for each level of temperature distinction between within and outside surface, 1 Watt of warmth vitality would course through each meter squared of its surface.

2.2. Preprocessing methods

The Box-Cox transformation and Principle component analysis are presented in the following. They not only scale the data but also change relationships between data points.

2.2.1. Box-Cox transformation

The Box-Cox transformation is a function, that is applied to the data, and is defined by a power parameter α. The aim is to transform the given data as close as possible to a normal distribution. It is defined as:

$$\begin{aligned} y_t^{(\lambda)} &= \frac{y_t^{\lambda} - 1}{\lambda}, \quad \lambda \neq 0, \\ y_t^{(\lambda)} &= ln(y_t), \quad \lambda = 0. \end{aligned} \tag{3}$$

Due to the strict positivity of input values, in this work, a constant with amount one is added to all data points before the transformation. The power parameter α is estimated using a maximum log likelihood estimation [16,17].

2.2.2. Principle component analysis (PCA)

PCA is a form of dimensionality reduction. It is used to reduce a high dimensional feature space into a lower dimension. In PCA, data from a high dimension n is projected onto a hyperplane which can be represented in $n-1$ dimensions. This hyperplane is chosen in a way that it retains the most variance along the input space. An example shown in Fig. 1 describes the idea for a 2-dimensional example projection on a 1-dimensional hyperplane.

X_1 shows the original data and Z_1 the preserved variance on the 1 d hyperplane, with hyperplane c_1 preserving the most variance. This works similar in higher dimensions and can be used to reduce the dimensionality arbitrarily. For further details please refer to Wold et al. [19].

2.3. Clustering algorithms

In the following section the clustering algorithms used in order to cluster the geometrical data are presented. To have a variety of differently working algorithms is important as they define the way a cluster is defined.

- The *k-Means algorithm* works with centroids. In the beginning of the algorithm, the cluster centroids are randomly assigned. Thereby the number of centroids *k* has to be assigned manually. After initializing the centroids, all points get assigned to the nearest cluster centroid. After that the centroid gets updated. This process is repeated, until the position of the centroids doesn't change anymore [18].
- The *agglomerative clustering* algorithm is formed hierarchically. Initially, each point starts in its own cluster. In every step, two clusters are merged according to a metric [20].
- The *DBSCAN* algorithm is a density based clustering algorithm. The number of clusters has not to be chosen beforehand. The algorithm works as follows: A point is defined as a core point, if more than a certain number of points *min_samples* are within a certain range epsilon = *eps* of distance to that core point; The distance can hereby be any metric. If a point is within a neighborhood of a core point, it belongs to the same cluster, that also holds for other core points. In that way, high dense regions are located. An instance that is not a core point or is within the range of a core point is considered noise [18].
- The *Shared nearest neighbor (SNN)* algorithm is an extension of the DBSCAN algorithm and tries to improve the inability of DBSCAN to assign cluster to regions of different density. First, a similarity graph of the data is constructed. For each data point a vertex is created in this graph. Two vertices in the graph get connected only if the two vertices are contained in each others closest k neighbor list. After that, the strength of each edge is computed. It refers to the shared neighbors in the neighbor list. If the neighbors of a vertex are not contained in the comparing vertex neighbor list, the vertices are not equal. If they have many neighbors in common, they are more equal. If this strength falls below a certain threshold, the edge is removed.

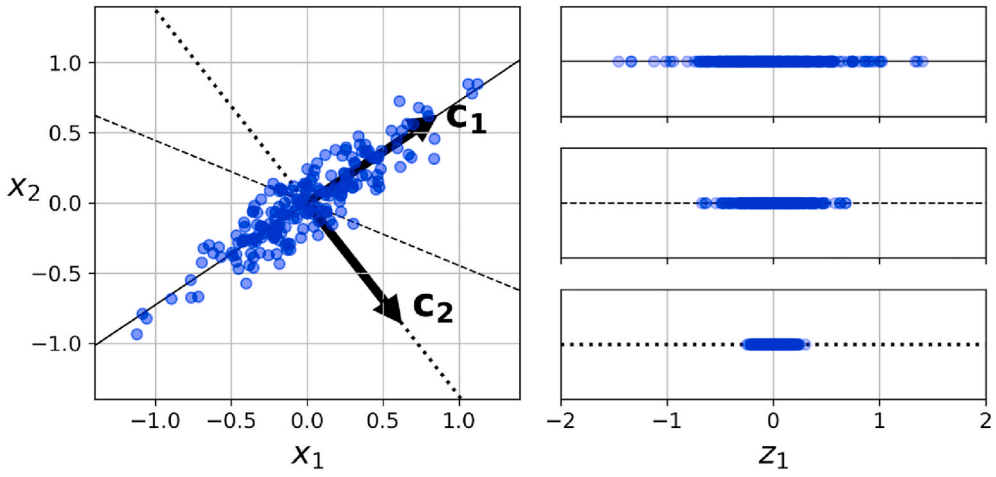


Fig. 1. Example for a dimensionality reduction with a Principle Component Analysis. Two dimensional data is reduced to one dimension, retaining as much variance as possible. In this example, the c_1 hyperplane of X_1 retains the most variance [18].

The connected components in the so formed graph are the assigned clusters. Vertices that are not connected, are denoted noise [21,22].

- *Mean shift* is a hill climbing algorithm that assigns clusters by shifting data points in each step towards high dense regions. Frequently used distance kernels are the Gaussian and Flat kernel.

The Flat kernel is defined as

$$F(v) = \begin{cases} 1, & \text{if } \|v\| \leq r \\ 0, & \text{if } \|v\| > r \end{cases} \tag{4}$$

and the unit Gaussian kernel as

$$G(v) = e^{\frac{-\|v\|^2}{2r^2}}, \tag{5}$$

where v is the regarded distance vector. In the following, the Mean shift algorithm with a Gaussian kernel is called **GMS** and with a Flat kernel **FMS**. In each step of the algorithm, the data points are shifted based on a proportion of the weighed summed distance vectors. These steps are repeated until convergence of all data points. A cluster consists of all data points that converge to the same point [23].

2.4. Clustering validation indices

The following indices represent a mathematical description of what a good clustering should look like. The calculations are taken from Van Craenendonck and Blockeel [24], which can also be referred to for further information.

- The *Davies-Bouldin (DB)* measure defines compactness based on the distance of each data point to the centroid of its cluster and separation on the distances of the cluster centroids:

$$S(c_k) = \frac{1}{|c_k|} \cdot \sum_{x_i \in c_k} d(x_i, \overline{c_k}), \tag{6}$$

$$DB(C) = \frac{1}{|C|} \sum_{c_k \in C} \max_{c_l \in C \setminus c_k} \left(\frac{S(c_k) + S(c_l)}{d(\overline{c_k}, \overline{c_l})} \right). \tag{7}$$

- The *Calinski-Harabasz (CH)* defines compactness based on the distance of each data point to the centroid of its cluster and separation on the distances of the cluster centroids to the data centroid $\overline{X}$,

$$CH(C) = \frac{(N - |C|) \cdot \sum_{c_k \in C} |c_k| d\left(\overline{c_k}, \overline{X}\right)}{(|C| - 1) \cdot \sum_{c_k \in C} \sum_{x_i \in c_k} d(x_i, \overline{c_k})}. \tag{8}$$

- The *Silhouette measure (SSC)* defines compactness based on the pairwise distances of data points in the cluster and separation based on the distances between all points in the cluster to all points in the nearest cluster.

$$s(x_i) = \frac{b(x_i) - a\left(x_i, c_j\right)}{\max\left(b(x_i), a\left(x_i, c_j\right)\right)} \tag{9}$$

$$SI(C) = \frac{1}{N} \sum_{i=1}^{N} s(x_i) \tag{10}$$

$a(x_i, c_j)$ is the average distance of x_i to all other points in its cluster c_j and $b(x_i)$ the minimum of average distances of x_i to the nearest cluster which is not c_j.

3. Method

Requirements for the applied method and the method itself are presented in this section. First, the data preprocessing steps are described, such as the different value calculations. Then, the metrics used for the evaluation are defined followed by an explanation of the applied method.

3.1. Data description

The dataset consists of 1027 sections from 75 commercial buildings. Sections are unique parts of a building, calibrated independently, and described by over 800 possible features.

Only a selection of these features can be used to perform the actual value prediction as, considering the data set size, the noise would dominate possible relationships. Moreover, similarities could be lost due to how buildings are modelled in the dataset. Each parameter f can be described trough $f = \{f_1, \ldots, f_n\}$, as some parameters have more than one instance, such as '"*Wall_1*", "*Wall_2*", …, "*Wall_n*". An instances is described by various values, for example the width, height, and U-value.

3.1.1. Surface calculation

The calculation of the surfaces and adjusted U-values is described through the following equations:

$$f_{A_{acc}} = \sum_{f_i \in f} width(f_i) \cdot height(f_i), \left[m^2\right] \tag{11}$$

$$f_{U_{acc}} = \sum_{f_i \in f} \frac{U(f_i) \cdot width(f_i) \cdot height(f_i)}{f_{A_{acc}}} \cdot \left[\frac{W}{m^2 \cdot K}\right] \tag{12}$$

The surfaces of all geometrical components are computed to overcome several problems. The modelling of the sections is not consistent trough the dataset. Moreover, calculating surfaces reduces dimensionality considerably, resulting in 25 surface values describing one section.

As an example, let S_1 and S_2 be two sections with equal windows W_1, W_2, W_3. The windows themselves are modelled through width, height, orientation and 12 other parameters. That already gives a feature space of 45 features, describing only the windows of the sections. Calculating and using only the surface reduces variance explained through the variables, also the dimensionality with only one remaining feature, the window surface. Despite that, the modelling differences are reduced through the calculation. Consider the modelling of windows of section $S_1 = \{W_{1_{s_1}} : W_1, W_{2_{s_1}} : W_2, W_{3_{s_1}} : W_3\}$ and $S_2 = \{W_{1_{s_2}} : W_2, W_{2_{s_2}} : W_1, W_{3_{s_2}} : W_3\}$. As instances of the same feature are compared, with such a modelling, the sections windows would not be equal. The overall surface is independent from such modelling differences, thus still equal.

3.1.2. Ratio calculation

All models in the dataset were modelled as having a rectangular prism form. In the following the calculation of the ratios, based on the surfaces of the geometrical components, is presented.

- The *floor area to zone volume ratio* (**FTZV**) describes the relationship between the overall floor area and the volume of the zone,

$$\textbf{FTZV} = \frac{Base\_Floor + Floor + Virtual\_Floor + Projected\_Floor}{Zone\_Volume}. \tag{13}$$

It is meant to differentiate between sections that have a different amount of stories. If a section has multiple stories, the FTZV will be higher than by a similar section with just one story. Similar hereby refers to the shape.

- The *window to wall ratio* (**WWR**) is used to define the proportion of glazed surfaces compared to envelope surfaces,

$$\mathbf{WWR} = \frac{Window + Box\_Window + Double\_Facade}{Base\_Wall + Wall}. \quad (14)$$

Thereby only *asymmetric* values are considered important for the computation, because they have more influence on the calibration [25].

- The *roof to floor ratio* (**RTFR**) describes the relationship between the roof and the floor area,

$$\mathbf{RTFR} = \frac{Roof\_Area}{Floor\_Area}. \quad (15)$$

Roof hereby means only *asymmetric* ceilings that are not made of glass. In this calculation, the floor area is the ground shape of the section, not the overall floor area. This value is therefor calculated from the zone volume and the zone height, which is possible due to the prism like shape of the sections.

- The *virtual roof to floor ratio* (**VRTFR**) is the proportion between the virtual roof and the floor area,

$$\mathbf{VRTFR} = \frac{Virtual\_Zone\_Ceiling + Gallery}{Floor\_Area}. \quad (16)$$

Sections that have a virtual ceiling are located inside the building and normally have no roof. Sometimes the virtual ceiling is also described through a gallery, if an HVAC component is placed inside it, like a heater mat.

- The *glass roof to floor ratio* (**GRTFR**) describes the ratio of Glass roof to the floor area,

$$\mathbf{GRTFR} = \frac{Section\_Glass\_Roof}{Floor\_Area}. \quad (17)$$

It is important to differentiate between sections that have a glass roof and sections that don't. Environmental impacts, like rain, sun and wind can have significantly more impact on a glass roof than on a normal roof.

- The *inner wall area to floor ratio* (**IWTFR**) describes the ratio of all inner walls to the ground surface of the floor,

$$\mathbf{IWTFR} = \frac{Partition\_Wall + Inner\_Wall}{Floor\_Area}. \quad (18)$$

This value is used to describe the inner thermal mass of a section. If a section has a high **IWTFR** it is able to maintain a given initial temperature for a longer time compared to a section that has a low value.

- The *shape factor* (**SF**) considers the shape of the building,

$$\mathbf{SF} = \frac{Envelop\_wall\_surfaces + 2 \cdot Floor\_Area}{Zone\_Volume} \quad \left[\frac{1}{m}\right] \quad (19)$$

Buildings with the same volume that differ in the shape, will have similar values. Fig. 2 shows an example; Buildings B and D have completely different shapes but the same shape factor.

3.2. Value range evaluation

The metrics used for the range evaluation are called application based metrics and are calculated based on the U-value feature space of the clustering outcome. In the following section, two methods for this evaluation are presented. The Mean range score focuses on differences in ranges between clusters whereas the Range percentage score evaluates the overall improvement of ranges. Data points that were assigned as noise by the clustering algorithm are not considered for evaluation of the metrics.

3.2.1. Mean range cluster score (MRCS)

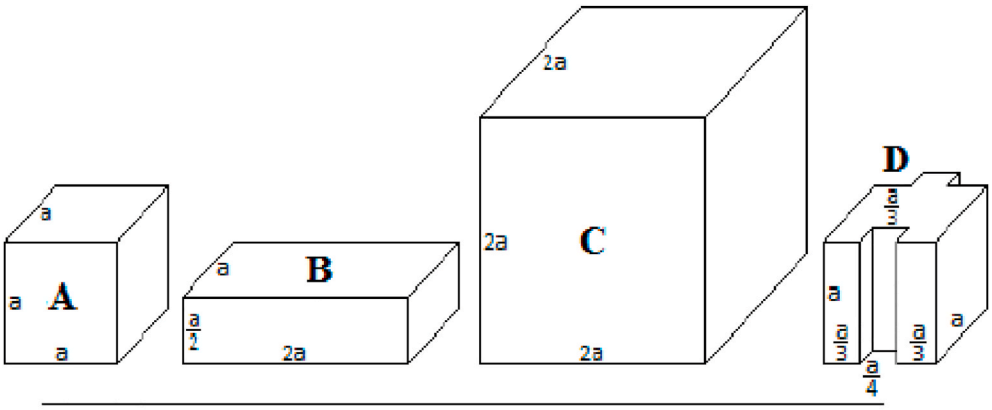
This metric describes the intersection between ranges of U-values from different clusters and a good score means that there is little overlapping. The aim is to check whether different clusters have different ranges, which would indicate that clustering benefits range definition. The score consists of 2 parts, the first part s_1 calculates the overlapping of the clusters and the second part s_2 the cluster structure itself.

The calculation for the joined score is:

$$s = s_1 + s_2. \quad (20)$$

The calculation of s_1 is the following:

$$x = \sum_{f \in F_U} \sum_{(C_i, C_j) \in C, i \neq j} intersect([Q_{0.05}(f(C_i)); Q_{0.95}(f(C_i))], [Q_{0.05}(f(C_j)); Q_{0.95}(f(C_j))]) \quad (21)$$



	A	B	C	D
Volume	a^3	a^3	$8a^3$	a^3
Thermal envelpe	$6a^2$	$7a^2$	$24a^2$	$7a^2$
Shape factor	6/a	7/a	3/a	7/a

Fig. 2. Motivation for the Shape factor, which is the thermal envelope divided by the volume of the section. Buildings B and D show very different shapes but have the same shape factor [26].

$$s_1 = \frac{1}{|F_U|} \cdot \frac{1}{\left|\left\{(C_i, C_j) \in C, i \neq j\right\}\right|} \cdot x \tag{22}$$

with $Q_{0.05}$ as the 5% percentile, respectively for 0.95 the 95%; $f(C_i)$ the value list of feature f for cluster C_i; F_U the U-value features of all surfaces; and *intersect* the area of intersection in percent between two ranges, calculated as the percentage of the smaller range.

The first part of the metric, s_1, describes how unique the thermodynamical value ranges are. Therefore, the average intersection between all cluster ranges is calculated. The intersection, in this context, is the percentage of the area of intersect regarding the smaller range. The percentiles of the ranges are considered to remove potential outliers.

It is possible that a feature is not represented in a cluster, then the thermodynamical value range is [0,0]. In such cases, these intersections are excluded from the calculation of the score.

The value of s_1 is between 0 and 1; the score of a clustering with very similar U-value ranges will be close to 1. Fig. 3 shows a case in which U-value ranges of clusters overlap.

In the example two ranges are presented: R_1: [1,5] and R_2: [4,6]. The overlapping is 1, which is 50% of the blue range R_2 and 25% of the red range R_1. The metric should describe the uniqueness of ranges, so the percentage of the smaller range R_2 is used.

The calculation of s_2 from (20) is the following:

$$\min \ = 0,02 \cdot |X|, \tag{23}$$

$$\max \ = 0,1 \cdot |X|, \tag{24}$$

$$s_2 = \sum_{i=0}^{|S|-1} \frac{|C_i|}{|X|} \cdot f(|C_i|)^2, \tag{25}$$

$$f(|C_i|) = \begin{cases} \dfrac{\min \ - (|C_i| - 1)}{\min} & \text{if } \ |C_i| < \min \ , \\ \dfrac{|C_i| - \max}{|X| - \max} & \text{if } \ |C_i| > \max \ , \\ 0 & \text{else} \end{cases} \tag{26}$$

s_2 evaluates the distribution of data points within a clustering. The boundaries min and max describe the desired range in which the average amount of data points for a cluster should be. The function $f(|C_i|)$ penalizes clusters that hurt these boundaries. The farer the assigned amount of data points is from the predefined range, the worse the score gets. Squaring $f(|C_i|)$ results in soft penalization close to the boundaries. The value of s_2 is between 0 and 1; 0 meaning all clusters are within the boundaries.

Let $C = \{C_1, C_2, C_3\}$ be a clustering with $|C_1| = 1000$, $|C_2| = 13$ and $|C_3| = 14$. s_1 can be very low for that clustering, as C_2 and C_3 could describe unique parameters that do not occur in C_1. However, such a clustering would not be desired for an initialization because new buildings are likely to get assigned to $|C_1|$. Little benefit would then be obtained from clustering the data.

3.2.2. Range percentage score (RPS)

The range percentage score describes the improvement of the clustered ranges compared to the initial ranges in percent.

$$\sum_{c_j \in C} \frac{|c_j|}{|X|} \cdot \frac{1}{|T|} \cdot \sum_{t_i \in T} \frac{\max\{x_k^{t_i} \mid x_k \in c_j\} - \min\{x_k^{t_i} \mid x_k \in c_j\}}{\max\{x_k^{t_i} \mid x_k \in X\} - \min\{x_k^{t_i} \mid x_k \in X\}} \tag{27}$$

The metric uses the average improvement over all clusters and thermodynamical values. The probability for a new section to get assigned to a big cluster is much higher than to get assigned to a small cluster. Therefore, smaller clusters have less impact compared to bigger clusters, as they would account for fewer range changes considering new clustered sections. Data points assigned as noise are considered for the initial ranges, but do not account for a cluster in the computation. T is the set of thermodynamical features, and $x_i^{t_i}$ for $t_j \in T$ describes the feature value t_j of x_i.

3.2.3. Mean sum of squared errors (MSSE)

The MSSE describes the average distance between data points and the centroid of their assigned cluster. It is calculated as

$$MSSE = \frac{1}{|X|} \cdot \sum_{c_i \in C} \sum_{x_i \in c_i} (x_i - \overline{c_i})^2 \tag{28}$$

3.3. Cluster models

Fig. 4 shows the structure of the applied method, In order to cluster the geometrical data and analyze the possibility of predictive clustering.

The input data first undergoes a basic Data cleaning procedure. Erroneous sections, such as dummy sections, are removed and wrong values in the dataset replaced. Then, the surfaces and ratios are calculated and all other features removed. After that, three different approaches are evaluated.

3.3.1. Approaches for clustering

The approaches differ in the feature space used for clustering. Approach 1 uses the calculated surfaces, with 25 features, for the subsequent procedure. For Approach 2 the ratios are considered, with 7 features. Approach 3 is a combination of both ratios and surfaces, resulting in 32 features. Each approach then proceeds either without transforming the data, or applying a Box-Cox transformation.

3.3.2. Box-Cox transformation

The majority of data distributions in the clustering dataset is right skewed, as often in real world data. There are many small values and only a few very big ones. Fig. 5 shows an example for the distribution of the window to wall ratio.

The issue occurring with such distributions, regarding clustering, is that distances vary over a broad range such that differences between sections with small values are neglected compared to big ones. The following figure illustrates the underlying problem for the window to wall ratio:

The window to wall ratio of section S_1 is 0, meaning it has no window surfaces. Compared to that, section S_2 has an equal amount of windows and walls. On the same scale, section S_3 and S_4 are more different than S_1 and S_2, but thats not an accurate representation of the reality. S_3 and S_4 both have significantly more windows than walls, which to a certain extend makes little difference. Compared to the difference of S_1 and S_2 their logical difference is almost negligible, but due to the skewness, they are less similar. The same holds not only for ratios but also surfaces with a similar explanation. Small sections have smaller surfaces, therefore differences between big sections are emphasized as the outweigh smaller differences.

The Box-Cox transformation, as a power transformation, is applied to overcome this problem of skewness. The benefit to single

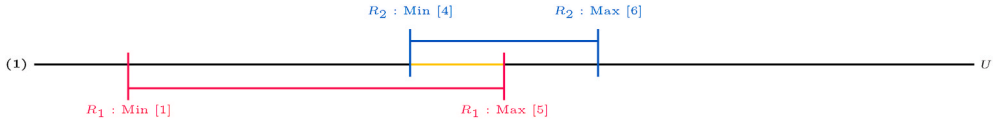


Fig. 3. Comparison of the intersection of two value ranges, marked in blue and red. The intersection, whose area is yellow, is 50%. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

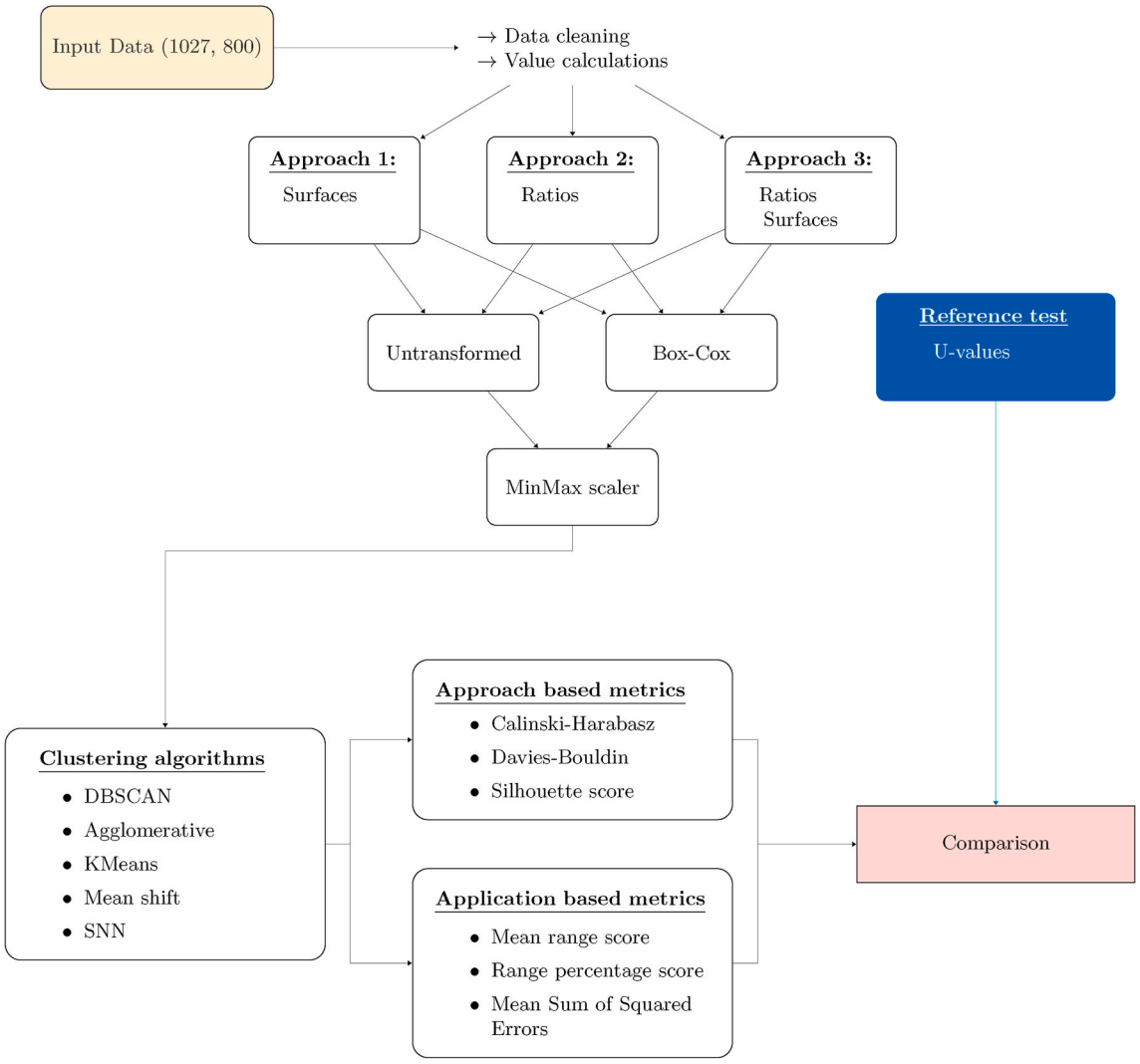


Fig. 4. Overview over the applied method.

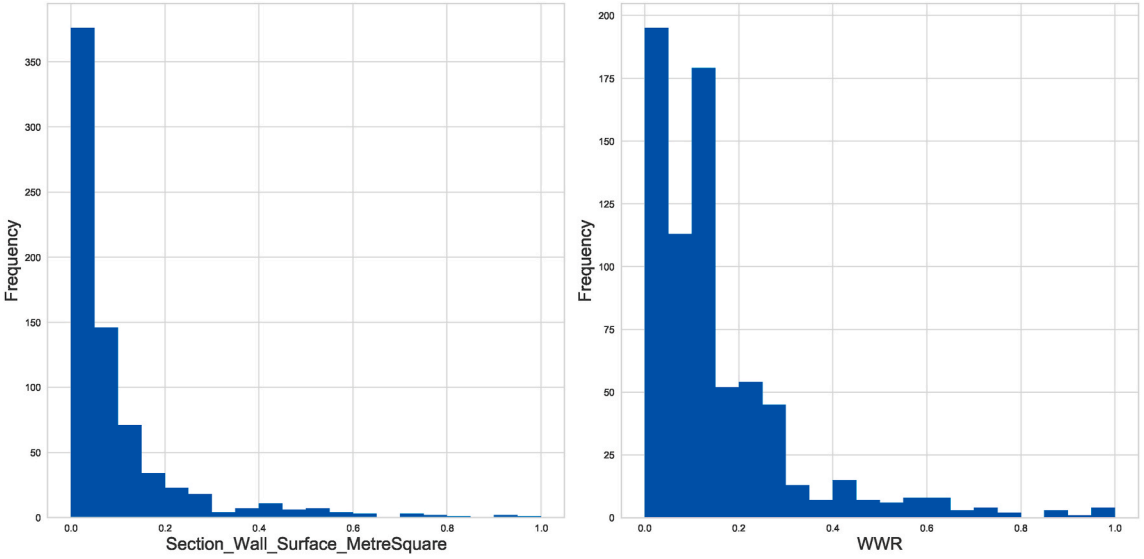


Fig. 5. Both plots show representative distributions of a parameter with the frequency on the y-axis and the corresponding value range on the x-axis. The window to wall ratio as well as the wall surfaces show a very right skewed distribution.

transformations, e.g. log-transforming, is, that it represents a broad range of possible transformers. The transformation is applied to each feature distinctly and the power parameter is estimated through a maximum likelihood estimation [17,27].

3.3.3. MinMax scaler

All features within an approach have different ranges of values. The surfaces and ratios of Approach 3, for example, all differ in their minimum and maximum value. In a clustering, especially with euclidean distance function, features are favored, whose ranges are smaller. The euclidean distance between two ratios, whose values normally are within [0, 6] is much smaller compared to the distance between two windows, ranging from 0 up to 1000. The data is, therefore, scaled to a (0, 1) range using a MinMax scaler to give every feature an equal weight for the clustering.

3.3.4. Clustering algorithms

The data is clustered using DBSCAN, Agglomerative, KMeans, Mean shift and shared nearest neighbor. The optimal amount of epsilon for DBSCAN is estimated trough a display of the knee plot [28,29]. The amount of clusters for KMeans is also determined trough a visual procedure, the elbow method, with the distortion score [30]. The same amount of cluster that is used for KMeans is also used for the agglomerative clustering. All other parameters are defined by testing.

3.3.5. Evaluation

There are two types of metrics to evaluate the clustering varying in the used evaluation feature space. The approach based metrics evaluate the geometrical similarity of a cluster by using the input data used for clustering. The application based metrics evaluate the clustering with the given aim of defining unique thermodynamical ranges. Therefore differences in the thermodynamical values of each cluster are evaluated. Finally, a comparison of all approaches and methods is performed. Results of transformation, geometrical clustering and thermodynamical range definition are evaluated. Moreover, a reference test is used, which traverses through the presented method using the U-values. It represents the optimum, concerning thermodynamical range definition.

4. Case study

In this section the results (see Table 1) of the case study for the applied method are presented. First, the impact of the Box-Cox transformation is investigated followed by an evaluation of the clustering algorithms based on the approach based metrics. Then, the relationship between the geometrical clusters and U-value ranges are investigated by the application based metrics.

Table 1
Results of the executed method. The approach based metrics *CH, DB, SSC* can only be compared within an approach and transformation. The application based metrics *MSSE, MRCS, RPS* should be compared between all approaches. The average score *AS* explains an average ranking of the application based metrics. For *SSC* and *CH* higher scores are desired, lower scores otherwise.

Approach	Transformation	Algorithm	Approach based metrics			Application based metrics			AS
			CH	DB	SSC	MSSE	MRCS	RPS	
Approach 1	BoxCoxTransforming	SNN	157.56	1.56	0.2	4.6	0.91	0.61	0.73
		MeanShiftGauss	84.78	0.69	0.37	5.18	1.05	0.7	0.89
		Agglomerative	305.19	1.14	0.39	5.13	0.97	0.74	0.86
		MeanShiftFlat	76.55	0.79	0.37	5.2	1.13	0.71	0.93
		DBSCAN	143.21	1.6	0.3	2.94	0.96	0.75	0.71
		k-Means	334.59	1.16	0.34	5.21	0.92	0.65	0.8
	None	SNN	40.69	1.85	0.02	3.02	0.89	0.45	0.54
		MeanShiftGauss	34.86	0.15	0.75	6.0	1.07	0.71	0.96
		Agglomerative	315.23	0.91	0.62	5.95	1.21	0.66	1.0
		MeanShiftFlat	158.35	0.53	0.57	5.7	1.0	0.64	0.87
		DBSCAN	1.0	1.0	1.0	1.0	1.0	1.0	0.72
		k-Means	326.47	0.88	0.62	5.96	1.21	0.66	1.0
Approach 2	BoxCoxTransforming	SNN	77.5	1.66	0.07	3.58	0.87	0.52	0.6
		MeanShiftGauss	99.47	0.72	0.31	5.47	1.23	0.55	0.92
		Agglomerative	249.7	1.44	0.22	5.45	0.97	0.54	0.79
		MeanShiftFlat	88.9	0.78	0.28	5.42	1.19	0.55	0.9
		DBSCAN	45.77	2.85	0.15	4.85	1.2	0.75	0.96
		k-Means	290.73	1.28	0.26	5.44	0.92	0.49	0.74
	None	SNN	56.38	1.66	0.06	3.99	0.93	0.49	0.64
		MeanShiftGauss	123.23	0.23	0.74	6.01	1.15	0.7	1.0
		Agglomerative	407.8	0.64	0.42	5.8	1.25	0.58	0.97
		MeanShiftFlat	253.19	0.41	0.5	5.74	1.15	0.62	0.94
		DBSCAN	1.0	1.0	1.0	1.0	1.0	1.0	0.72
		k-Means	453.97	0.57	0.49	5.78	1.27	0.6	0.99
Approach 3	BoxCoxTransforming	SNN	92.69	1.68	0.17	4.43	0.9	0.62	0.72
		MeanShiftGauss	38.57	0.72	0.21	5.03	1.04	0.6	0.83
		Agglomerative	205.62	1.35	0.26	5.19	0.97	0.7	0.85
		MeanShiftFlat	48.57	1.01	0.17	5.09	1.12	0.69	0.91
		DBSCAN	88.05	1.63	0.12	1.97	0.83	0.65	0.53
		k-Means	220.4	1.3	0.31	5.07	0.94	0.69	0.82
	None	SNN	52.32	2.05	−0.09	3.74	0.91	0.46	0.6
		MeanShiftGauss	28.99	0.17	0.66	5.96	1.01	0.7	0.92
		Agglomerative	198.24	0.9	0.5	5.9	1.21	0.66	1.0
		MeanShiftFlat	89.66	0.55	0.44	5.66	1.02	0.62	0.87
		DBSCAN	1.0	1.0	1.0	1.0	1.0	1.0	0.72
		k-Means	212.44	0.93	0.37	5.88	1.22	0.62	0.98
Reference	None	SNN	165.21	1.63	0.25	0.74	0.6	0.18	0.11
		MeanShiftGauss	102.89	0.38	0.38	0.65	0.8	0.19	0.2
		Agglomerative	387.66	1.21	0.36	2.06	0.68	0.33	0.31
		MeanShiftFlat	103.43	0.69	0.32	1.15	0.83	0.27	0.29
		DBSCAN	123.71	1.48	0.23	0.17	0.48	0.17	0.0
		k-Means	431.3	1.2	0.34	1.91	0.65	0.3	0.27

4.1. Comparison between different clustering and preprocessing methods

The application of different transformation methods has a significant impact on the outcome of the clustering. The results without applying a transformation are analyzed first, followed by applying a Box-Cox transformation.

4.1.1. No transformation

Without the transformation, the sections are scattered unproportionally over a large range; one dense region with many very close data points and widely spread data points otherwise. Fig. 6 illustrates this exemplary for all non-transformed approaches.

Two main reasons causing the distribution can be outlined:

1. The inconstant occurrence of features for sections. Some features are more likely to occur in a section than others; all sections have walls but only a small percentage have a glass roof. The distances of sections that do not have a feature to sections that do have is, naturally, bigger compared to two sections that both have this feature.
2. The most often right skewed parameter distribution over the dataset. The majority of the features, if they occur, have a small value. Considering that the distance is measured with the L2 norm, the distance between two small sections, despite real differences, is considerably smaller than the distance between two big sections.

This distribution itself is just a representation of the data set. However, a problem can occur when using common clustering methods. When applying DBSCAN, MSG, or MSF, the result is, for all non-transformed approaches, one very dense cluster with the majority of data points and small clusters otherwise. These algorithms are designed to find dense regions in a feature space, presumed dense regions are defined similarly in the sense of the metric they use. All of the three algorithms have a constant parameter, the epsilon is constant in DBSCAN and the Mean shift algorithm has a constant bandwidth. These constant parameters make it impossible to cluster such skewed feature distributions. Varying the parameters has little impact on the outcome; decreasing it results in countless clusters of only one data point and an increase in only one cluster.

k-Means and Agglomerative determine more evenly distributed clusters, but still one cluster containing the majority of the data points (70%). *k*-Means uses centroids for assigning data points to clusters, the average distance of data points to their centroid is minimized. In the recomputation of a centroid, sections with unique features or a considerably big size have a significant impact. Such sections account for a greater distance to the assigned centroid, weighing more in the recomputation. Smaller, more similar sections account for less variance and weigh less in a recomputation of a centroid, leading to assigning more to only one cluster. This is similar for Agglomerative clustering, which uses pairwise distances to centroids. Small sections are more similar compared to big sections, thus more likely to get merged in the clustering process.

Only the shared nearest neighbor algorithm is designed to determine clusters of different densities, as it considers the number of joined nearest neighbors of a data point. It determines, different to the preceding algorithms, clusters of a more similar size. However, SNN assigns similar distributed, but also many clusters. The clusters contain between 5 and 10% of the data points, with 20–25 clusters.

4.1.2. Box-Cox transformation

Applying the Box-Cox transformation has a considerable impact on the distribution of the feature spaces. All algorithms, despite from SNN, benefit from the transformation in means of a more even clustering distribution. But also, as discussed later, in the application based metrics. The initially very skewed distribution is unskewed by the transformation to a more comparable scale. Fig. 7 shows the impact of the transformation on an example distribution.

As discussed in the preceding chapter, the right skewed distribution leads to an skewed distributed feature space and uneven cluster distributions. The transformation leads to a more similar scale, which benefits the algorithms. Table 2 shows an example result of the Box-Cox transformation applied to wall surfaces. Before the transformation, the distances between S_1 and S_2, $D(f_1(S_1), f_1(S_2)) = 10.61$, were only 10% compared to the distance between S_3 and S_4, $D(f_1(S_3), f_1(S_4)) = 101.79$. After the transformation, the distances are on an equal scale and comparable. The high distance between sections S_1, S_2 and S_3, S_4 remains.

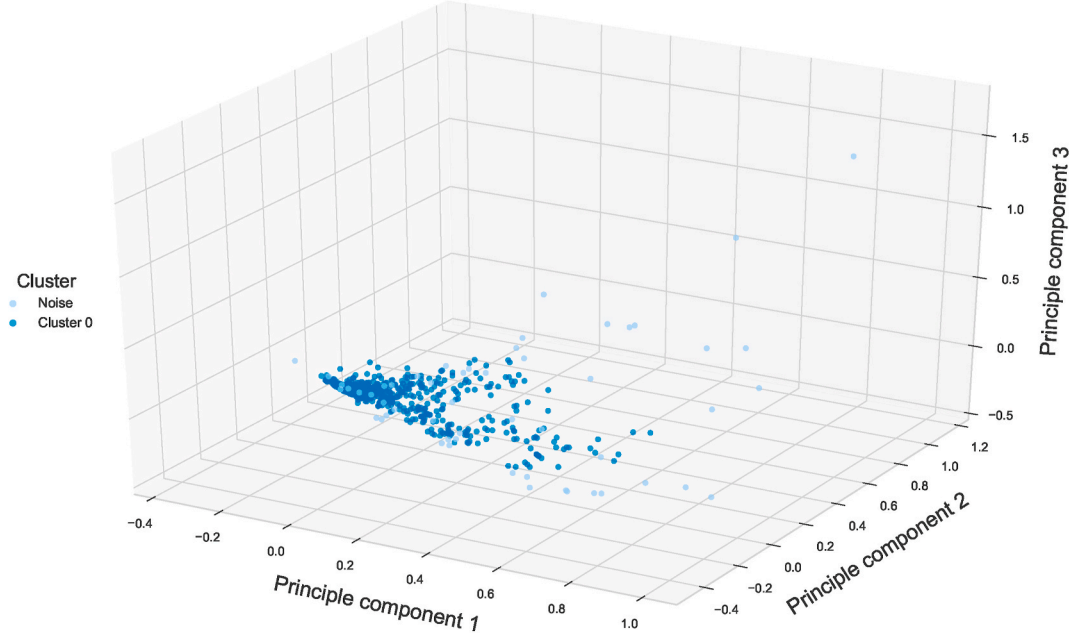


Fig. 6. Feature space distribution of Approach 3, without a transformation. The colors of the data points refer to the assigned cluster by the DBSCAN algorithm. The feature spaces' dimensionality is reduced using a PCA reduction, the remaining variances are [38.2%, 16.9%, 10.1%]. Data points with the label −1 are considered as noise by the algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

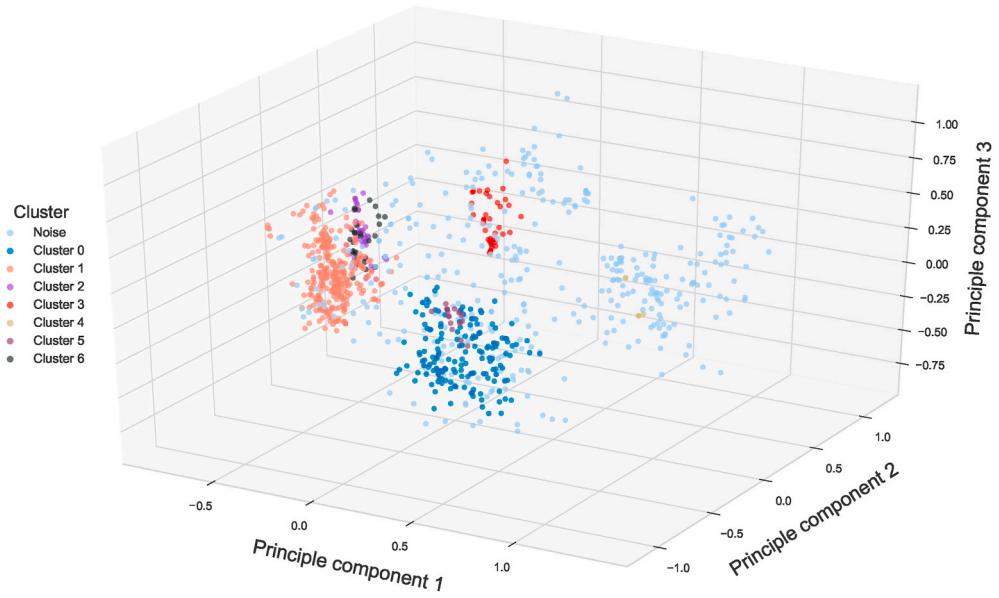


Fig. 7. Feature space distribution of Approach 3, with applied Box-Cox transformation. The colors of the data points refer to the assigned cluster by the DBSCAN algorithm. The feature spaces' dimensionality is reduced using a PCA reduction, the remaining variances are [28.4%, 20.8%, 10.2%]. Data points with the label −1 are considered as noise by the algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Table 2
Result of applying the Box-Cox transformation with $\alpha = 0.1075$ on different wall surfaces of the data set. A constant of 1 is added preceding the Box-Cox calculation, as the data has to be strictly positive.

	Non-transformed	Transformed
$f_1(S_1)$	100.64	5.99
$f_1(S_2)$	111.25	6.15
$f_1(S_3)$	1000.71	10.25
$f_1(S_4)$	1102.50	10.45

4.2. Evaluation of geometrical clusters using clustering validation indices

A comparison between the approach based metrics can only be performed within one feature space. For example, the results of Approach 1 with a Box-Cox transformation can not be compared to the results of Approach 1 without a transformation. This is due to the computation of the metrics, for further information on the computation please refer to Van Craenendonck and Blockeel [24]. Therefore, in this chapter, the average performance of an algorithm in the different feature spaces is evaluated.

4.2.1. Differences in algorithms

DBSCAN and SNN account for the worst scores in all approaches and approach based metrics. This is the reason for the approach based metrics being hardly able to evaluate distributions other than spherical [24].

The Mean Shift algorithm, with both kernels, is among the worst Calinski-Harabasz scores in all feature spaces. In the three non-transformed approaches, the algorithm assigns, due to the constant bandwidth, one cluster with 98% of the data points and single point clusters otherwise. As the centroid of the main cluster is almost equal to the data centroid, the Calinski-Harabasz score is low.

The behavior of the algorithm does not change in the Box-Cox transformed versions of the approaches, a small number of very big cluster and smaller clusters otherwise are determined. However, this behavior benefits the Davies Bouldin score. Mean shift assigns close dense regions into one cluster, which reduces scaterdness between nearest clusters; scattered near clusters are merged into one. Thus, the Davies-Bouldin score improves. The Silhouette score also benefits from the reduced scaterdness, but mostly in the non-transformed approaches; more single point clusters get assigned and one big dense cluster, defining a clear cluster belonging.

Compared to that, *k*-Means and Agglomerative achieve the overall best Calinski-Harabasz scores but low Davies-Bouldin and Silhoutte scores in the non-transformed feature spaces. This can be explained by how these algorithms define clusters, which is very similar.

k-Means initializes centroids with a probability distribution over the feature space and the centroids are shifted based on the assigned data points. Agglomerative merges clusters, whose centroids have the smallest distance, hierarchically into one cluster. For non-transformed approaches, this results in splitting the dense region into multiple clusters, as the number of data points in the dense region outweigh the distance to data points in sparse regions. Fig. 8 illustrates that splitting for the Agglomerative algorithm in Approach 1.

Splitting the dense region results in a bad Silhouette and Davies Bouldin score for the non-transformed approaches. One dense region is divided into two, so the clusters are not clearly defined, near, and data points between the clusters have no definite cluster belonging. However, a better Calinski-Harabasz score is achieved because, through the splitting, the data centroid is less equal to the clusters centroids.

k-Means and Agglomerative perform almost as good in the Silhouette score after the transformation because dense regions are more scattered among the feature space, making it less likely for a dense region to be divided. The Davies Bouldin score remains bad compared to the Mean shift approaches, as they are designed to assign scattered regions into one cluster work differently compared to the preceding algorithms, as they are density based. Their procedure does not involve a distance measure to a centroid, which is the result of having a bad score in all Approach based metrics.

4.2.2. Algorithms and geometrical clusters

The Mean shift algorithm, independent of the kernel, assigns near, scattered regions into one cluster and often unique sections into one single cluster.

For assigning different building patterns, this is not beneficial. Having few clusters with the majority of the data points and one point cluster otherwise does not contain information regarding patterns of different sections. Especially, considering that other algorithms are able to determine some patterns.

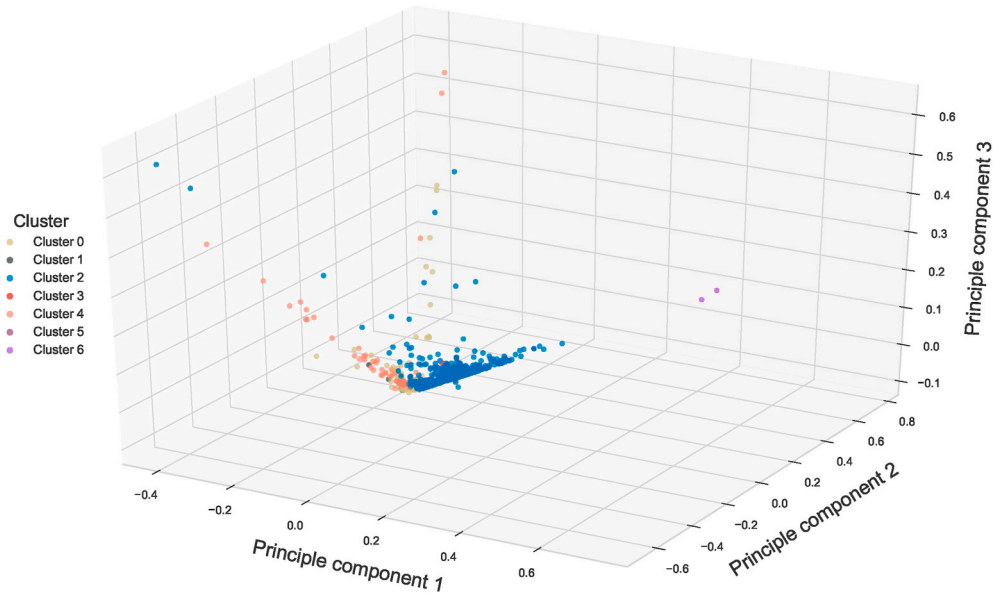


Fig. 8. Feature space distribution of Approach 1, without a transformation. The colors of the data points refer to the assigned cluster by the Agglomerative algorithm. The feature spaces' dimensionality is reduced using a PCA reduction, the remaining variances are [35.0%, 21.3%, 17.1%]. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

k-Means and Agglomerative assign the clusters based on the Euclidean distance. DBSCAN computes the distance for epsilon on the l2 norm, too. This seem to not be representative in the non-transformed approach, as real world differences hardly can be represented. Two sections with small values can be more different compared to two sections with big values, but as differences are measured Euclidean the differences between big sections will always have a greater distance, as the example in Table 2. In the transformed approaches, however, these algorithms seem to define clusters better. Table 3 shows the evaluation of clusters of a clustering of *k*-Means for Approach 3 with a Box-Cox transformed feature space.

The differences in sizes are reduced through the Box-Cox transformation. Now, the algorithms assigns the clusters based on unique parameters, as it seems distances between combinations of occurring parameter outweigh potential smaller size differences. Agglomerative and even DBSCAN assigned very similar clusters as the example of *k*-Means, varying mostly in the belonging of single points. Mean shift with a Gaussian kernel merges cluster 1, 3, and 5 into one cluster. Mean shift with a flat kernel, compared to that, merged cluster 1,3 and 2,5. Considering the differences defining the clusters of the *k*-Means example, the Mean shift algorithm, independent from kernel, seems to not be appropriate for assigning geometrical clusters.

SNN further divides the clusters, shown in Table 3 into sub clusters, having an overall clustering amount of 14. SNN is the only algorithm that gives interpretable results for a non-transformed approach. It assigns the clusters based on the size of a section and the occurrence of unique parameters. This results in defining more clusters compared to the other approaches. By analyzing the defined clusters it turned out that SNN often assigns sections of the same building into the same cluster, which is, as they are geometrically very similar, a desired behavior. However, by assigning so many clusters SNN generalizes not well.

Table 3
Description of geometrical clusters defined by the *k*-Means algorithm, Approach 3, Box-Cox transformation. The main factor of assigning a cluster is whether a parameter occurs in a cluster or not.

i	C_i	Unique parameters of the cluster	Function of the cluster
0	104	Low WWR, no roof	Sections in the center of a building
1	244	Roof, high floor and wall surfaces	Sections on top of a building
2	172	Base floor and roof	Sections that have no other section below or on top of them
3	302	no roofs, high WWR	Sections in the middle of a building with big windows
4	74	Sections with a high IWTFR	Sections with many walls inside
5	106	No roof, base floor	Sections on the bottom of buildings
6	25	Glass roof	Atriums, entry halls

4.3. Relationship between geometrical clusters and U-value ranges

The application based metrics measure the relationship between the geometrical clusters and the U-value ranges. The data is clustered on the geometrical data but evaluated on the thermodynamical. The application based metrics can be, compared to the approach based metrics, compared between clustering feature spaces.

The overall best average score of 0.53 is achieved by DBSCAN of Approach 3 with a Box-Cox transformed feature space. This is closely followed by the SNN algorithm applied to the non-transformed approaches, with SNN of Approach 1 having an average score of 0.54. Both, DBSCAN and SNN assign data points as noise. DBSCAN, compared to SNN, assigns almost twice as much data points as noise. 374 Data points in DBSCAN approach and 189 in SNN. SNN in Approach 3 without a transformation is the third best performing approach, scoring 0.6, with only 74 assigned data points as noise. Considering that DBSCAN assigns much more noise than SNN, the score is only slightly better.

DBSCAN has a better MSSE and MRCS, whereas SNN account for a better RPS. This could seem contradictive, reckoning that the MSSE measures the distance to the centroid and the RPS measures the overall range. Having a small RPS should normally correlate with a good MSSE. However, as both scores do not take the noise into account, the MSSE can be low because the assigned cluster account for overall smaller ranges, resulting in a small MSSE. The RSP is computed as the improvement of an assigned clustering value range compared to not clustering the data; therefore, data points considered as noise are not taken into account for the initial comparing value ranges but only the overall cluster ranges themselves. Having a better MSSE but worse RPS indicates, that the ranges of DBSCAN are smaller on average, but the improvement obtained to not clustering the data is bigger for SNN. That

leads to the assumption that the assigned noise has a big impact on the result of DBSCAN, as the clustering itself does not benefit as much.

The MRCS is better in DBSCAN, meaning the ranges intersect less. This leads to the conclusion that the noise assigning of DBSCAN assigns Data points, which account for bigger ranges in SNN as noise, so that SNN can have a better improvement on ranges by computing on a bigger range, but smaller value ranges for DBSCAN.

The best performing algorithms of the approach based metrics are comparably bad. The best score for *k*-Means is 0.74; Agglomerative, 0.79; and MSG, 0.83. Possible reasons for these differences are analyzed in the following chapter. The first difference is that the best algorithms both assign noise, so trying to assign noise for the corresponding algorithms is a first analysis approach. Further, for *k*-Means and Agglomerative the number of clusters is predefined, for DBSCAN and SNN not. Assigning a higher number of cluster is therefore also analyzed.

In the following it is first determined how significant the results of the best performing approaches DBSCAN and SNN are. After that, possible reasons for the better performance of DBSCAN and SNN are analyzed. They obtained the best average scores for the application based metrics, even though the obtained the worst in the approach based metrics.

4.3.1. Significance test

To first have an impression of the significance of the test results, a significance test was performed for the two best performing approaches, SNN of Approach 1 without a transformation and DBSCAN of Approach 3 with a Box-Cox transformation.

The question to answer is whether the result of the metrics are a result of randomly distributed ranges across the cluster. The null hypothesis tested is "*The scores of the metrics are the result of randomly distributed thermodynamical ranges*". To test the hypothesis, the labels were reassigned randomly, the metrics calculated again, and counted how often the score is better than the initial one. The reassigning was performed 300 times for both approaches. Table 4 shows the results. The p-value is in all cases close to zero, which leads to a rejection of the null hypothesis. This makes it likely that the scores are not the result of randomly distributed ranges.

4.3.2. Noise assigning

SNN and DBSCAN both assign noise, compared to *k*-Means, Agglomerative or Mean shift, which do not. To analyze the impact of the noise assigning, the method is executed again for these algorithms. This time, before executing the clustering, the data points considered as noise for SNN and DBSCAN were removed for the other algorithms within their approach. This can only be done within one feature space, as the feature space effects the noise assigning.

DBSCAN still shows a better result in most cases, though not all. Table 5 shows the results. The MSSE of DBSCAN is much better compared to the other algorithms, but MSG and SNN have a better MRCS and SNN a better RPS, too. That indicates, that the noise assigning has a considerable impact on the result of DBSCAN.

This is different for SNN, shown by Table 6. SNN still outperforms all other algorithms by far. This underlines the assumption that SNN by itself performs better when clustering building data, given the aim of obtaining thermodynamical value ranges.

Table 4
Results of testing the Null-hypothesis "*The scores of the metrics are the result of randomly distributed thermodynamical ranges*" for DBSCAN of Approach 3 with a Box-Cox transformation and SNN of Approach 1 without applying a transformation. How often the randomly assigned labels account for a better result in the metrics are displayed below the algorithm names.

Metric	Reassignments	DBSCAN	SNN
MSSE	300	0	0
MRCS	300	0	0
RPS	300	4	0

Table 5
Results of the re executed method for Approach 3, Box-Cox transformation. Data points assigned as noise by DBSCAN were removed from the evaluation space and the metric results computed again for the other algorithms of the approach.

Approach	Transformation	Algorithm	Application based metrics		
			MSSE	MRCS	RPS
Approach 3	BoxCoxTransforming	DBSCAN	1.97	0.83	0.65
		SNN	2.59	0.82	0.52
		MeanShiftGauss	3.29	0.79	0.53
		Agglomerative	3.27	0.8	0.65
		MeanShiftFlat	4.64	1.13	0.66
		k-Means	3.13	0.83	0.66

Table 6
Results of the re executed method for Approach 1, without a transformation. Data points assigned as noise by SNN were removed from the evaluation space and the metric results computed again for the other algorithms of the approach.

Approach	Transformation	Algorithm	Application based metrics		
			MSSE	MRCS	RPS
Approach 1	None	SNN	3.02	0.89	0.45
		MeanShiftGauss	6.21	1.24	0.74
		Agglomerative	6.15	1.22	0.67
		MeanShiftFlat	6.03	1.08	0.66
		DBSCAN	1.0	1.0	1.0
		k-Means	6.15	1.21	0.67

4.3.3. Number of clusters

For the SNN algorithm, the number of clusters could be relevant for achieving the best scores. It assigns disproportionately many, in the best performing approach 23. *k*-Means and Agglomerative are the only algorithms with a predefined number of clusters. To analyze the impact of assigning a high number of clusters, the algorithms were executed again with the same amount of clusters as SNN defines. That means, for example, *k*-Means of Approach 1 without a transformation was executed again with the number of clusters defined by SNN, which in this case, was 23. The results are shown in Table 7. The Box-Cox transformed versions of *k*-Means and Agglomerative improved, but are still worse than SNN. The Agglomerative of Box-Cox transformed Approach 2 improved considerably and even outperforms the SNN in the MRCS.

The number of clusters has a strong impact on the score outcomes, but can not be singled out as the only reason SNN outperforms the other algorithms.

4.3.4. Reference test

The reference test was included to have comparable values of an achievable optimum. All average scores of the reference test are better than the best score of the approaches. The best score of the approaches was 0.53, this is less than half as good as the best score of the reference test. In all three scores the reference outperforms the approaches, even the worst performing algorithm on the reference test has a better score. However, considering that the data was clustered on geometrical data the scores of the approaches are comparably good. The geometrical data in itself has no influence on the thermodynamical characteristics of a feature. Considering this, some of the approaches are very good in obtaining thermodynamical value ranges from geometrical data.

4.4. Discussion of the results

The results of the executed method were evaluated from two perspectives. The possibility of defining geometrical patterns and the predictive ability of such patterns, whether it is possible to obtain unique U-value ranges.

Table 7
Results of the re executed method for all approaches, whereas for each feature space, the number of Clusters for *k*-Means and Agglomerative got assigned as the number of clusters determined by SNN. The AS is comparable within the table as the average result.

Approach	Transformation	Algorithm	Application based metrics			AS
			MSSE	MRCS	RPS	
Approach 1	BoxCoxTransforming	SNN	4.6	0.91	0.61	0.57
		Agglomerative	4.51	0.93	0.6	0.56
		k-Means	4.69	0.93	0.62	0.62
	None	SNN	3.02	0.89	0.45	0.0
		Agglomerative	5.71	1.1	0.58	0.86
		k-Means	5.7	1.1	0.57	0.83
Approach 2	BoxCoxTransforming	SNN	3.58	0.87	0.52	0.19
		Agglomerative	4.02	0.88	0.47	0.17
		k-Means	4.64	0.94	0.49	0.36
	None	SNN	3.99	0.93	0.49	0.27
		Agglomerative	5.8	1.2	0.59	1.0
		k-Means	5.78	1.19	0.58	0.97
Approach 3	BoxCoxTransforming	SNN	4.43	0.9	0.62	0.56
		Agglomerative	4.54	0.95	0.62	0.63
		k-Means	4.79	0.95	0.6	0.63
	None	SNN	3.74	0.91	0.46	0.13
		Agglomerative	5.81	1.17	0.61	1.0
		k-Means	5.79	1.18	0.6	0.99

The MSG, *k*-Means and Agglomerative had on average the best results in the approach based metrics, which were meant to measure geometrical belonging. These metrics, however, showed to not be able to evaluate geometrical clusters given the defined feature spaces. Uneven cluster distributions achieved better scores, even though they assigned completely different geometrical patterns into one cluster.

The Euclidean distance was used throughout as a distance measure. Without applying a transformation, this turned out to not represent real differences accurately. Only the SNN algorithm was able to use the Euclidean distance in a non-transformed approach with good results. It was able to determine geometrical patterns and assign similar sections into clusters. Sections of the same building often ended up in the same cluster, accounting for the majority of data points. This definitely is clustering by geometrical patterns; however, it also leads to a lack of generalization, which can be a problem by using the resulting U-value ranges.

Applying a transformation leads to significantly better results regarding geometrical belonging, as Table 3 illustrated. The feature spaces are distributed into multiple dense regions instead of only one. Contradicting the initial guess, the dense regions are mainly determined by the occurrence of a feature in a section and not by the size of the section. The dense regions, and clusters, are defined mostly by unique parameter combinations and not sizes of sections.

Having a good approach based metric score was no indication for also achieving a good application based metric score. SNN and DBSCAN had the best application based metrics scores but the worst approach based. There are two possible explanations for this, either the approach based metrics are not good for evaluating geometrical patterns, or SNN and DBSCAN do not cluster geometrically similar sections. Considering that, at least, SNN was able to assign clusters based on building belonging, the ability of determining geometrical patterns seem to be given. The approach based metrics seem to not be able to evaluate a geometrical belonging, also in the transformed approaches.

Further test showed that DBSCAN only performed better than the other algorithms because it assigned a major part of the input data as noise. Compared to that, SNN worked better than the other algorithms in defining unique thermodynamical value ranges, even though it defines many clusters and noise. In the following tests, the other algorithms were first executed with the noise removed and then with the same amount of clusters determined by SNN. They still performed worse than the SNN algorithm.

Compared to the reference, the scores of the approaches were low. The worst performing algorithm of the reference approach was still significantly better than the best performing approach. However, considering that the ranges were determined by using the geometrical data only, the scores are surprisingly high. Compared to not using clustering, the SNN algorithm was able to reduce the value ranges on average by 50%.

5. Conclusion and future work

The metrics, Calinski-Harabasz, Davies-Bouldin, and Silhouette score, used for evaluating the geometrical clustering were not able to determine geometrical pattern belonging. Algorithms assigning sections independent of their geometrical properties accounted for the best scores.

The mean shift algorithm was, independent of the kernel used, not able to bring satisfying results. Neither was it able to determine building patterns, nor to define distinct U-value ranges, having a best average score of only 0.83.

Applying the Box-Cox transformation lead to overall better results compared to not applying the transformation. After applying the transformation, the sizes of sections were more similar, whereas then the occurrences of parameters accounted for the most variance.

k-Means, Agglomerative and DBSCAN were not able to determine anything other than huge size differences in the non-transformed feature space, having a best average score of 0.97. Only after the transformation other patterns were determined, improving the best average score to 0.53.

DBSCAN with Box-Cox transformed surfaces and ratios accounted for the best thermodynamical range scores. However, an analysis showed that this was mainly the reason of assigning over 1/3 of the data points as noise.

The overall best approach is SNN, as it determines geometrical patterns best and accounts for the most unique thermodynamical value ranges with a best average score of 0.54. It also assigns noise, but much less. Also, this showed to not be the reason for SNN performing better. The U-value ranges were on average 50% of the initial value ranges, if the whole available feature ranges were used. If no noise should be assigned, the *k*-Means algorithm showed the best results with applying a Box-Cox transformation on the ratio values.

The first thing that should be investigated in the future is the usability of the U-value predictions. The ranges should be tested with real world buildings; a set of test buildings is initialized with predicted U-value ranges, then it should be compared how the calibration works with and without using the range prediction.

It should further be tested whether the inputs could be defined in a way, that produces better results. Something else than the geometrical surfaces could be calculated or other ratios. In addition to this, some other parameters could be made available, like the last buildings renovation year, which could also give an indication of the thermodynamical building parameters.

Other clustering algorithms in combination with different metrics should be used in order to check whether they produce better results. Metrics could be the cosine metric, which measures the angle between two data points therefore if these point in the same direction.

Other methods worth investigating are supervised and semi-supervised approaches, because for existing buildings the U-values are already available. The challenging part is, that these are multi dimensional continuous outputs, so maybe a neuronal network could be used.

Credit author statement

Simon Klemp: Conceptualization, Methodology, Software Writing - Original Draft, Visualization.

Ahmed Abida: Conceptualization, Methodology, Software Writing - Original Draft, Visualization.

Pascal Richter: Conceptualization, Methodology, Writing - Original Draft, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the German Federal Ministry for Economic Affairs and Energy for funding and possibility to realize the project “DEOKI” in frame of the program “Zentrales Innovationsprogramm Mittelstand” (ZIM, Project ZF4632102HM9).

References

[1] C.H. Antunes, V. Rasouli, M.J. Alves, Á. Gomes, J.J. Costa, A. Gaspar, A discussion of mixed integer linear programming models of thermostatic loads in demand response, in: International Symposium on Energy System Optimization, vols. 105–122, Birkhäuser, Cham, 2018.

[2] L. Yang, H. Yan, J.C. Lam, Thermal comfort and building energy consumption implications–a review, Appl. Energy 115 (2014) 164–173.

[3] C. Zhu, W. Tian, B. Yin, Z. Li, J. Shi, Uncertainty calibration of building energy models by combining approximate Bayesian computation and machine learning algorithms, Appl. Energy 268 (2020) 115025.

[4] M. Gangolells, M. Casals, J. Ferré-Bigorra, N. Forcada, M. Macarulla, K. Gaspar, B. Tejedor, Office representatives for cost-optimal energy retrofitting analysis: a novel approach using cluster analysis of energy performance certificate databases, Energy Build. 206 (2020) 109557.

[5] T.G. Nikolaou, D.S. Kolokotsa, G.S. Stavrakakis, I.D. Skias, On the application of clustering techniques for office buildings’ energy and thermal comfort classification, IEEE Trans. Smart Grid 3 (4) (2012) 2196–2210.

[6] L. Filogamo, G. Peri, G. Rizzo, A. Giaccone, On the classification of large residential buildings stocks by sample typologies for energy planning purposes, Appl. Energy 135 (2014) 825–835.

[7] A. Satre-Meloy, M. Diakonova, P. Grünewald, Cluster analysis and prediction of residential peak demand profiles using occupant activity data, Appl. Energy 260 (2020) 114246.

[8] R. Hecht, G. Meinel, M. Buchroithner, Automatic identification of building types based on topographic databases–a comparison of different data sources, Int. J. Cartogr. 1 (1) (2015) 18–31.

[9] G. Tardioli, R. Kerrigan, M. Oates, J. O’Donnell, D.P. Finn, Identification of representative buildings and building groups in urban datasets using a novel pre-processing, classification, clustering and predictive modelling approach, Build. Environ. 140 (2018) 90–106.

[10] S. Zuhaib, M. Hajdukiewicz, J. Goggins, Application of a staged automated calibration methodology to a partially-retrofitted university building energy model, J. Build. Eng. 26 (2019) 100866.

[11] A. Chong, K.P. Lam, M. Pozzi, J. Yang, Bayesian calibration of building energy models with large datasets, Energy Build. 154 (2017) 343–355.

[12] M.H. Kristensen, R.E. Hedegaard, S. Petersen, Hierarchical calibration of archetypes for urban building energy modeling, Energy Build. 175 (2018) 219–234.

[13] A. Chong, K. Menberg, Guidelines for the Bayesian calibration of building energy models, Energy Build. 174 (2018) 527–547.

[14] H. Lim, Z.J. Zhai, Influences of energy data on Bayesian calibration of building energy model, Appl. Energy 231 (2018) 686–698.

[15] Z. Shi, W. O’Brien, Sequential state prediction and parameter estimation with constrained dual extended Kalman filter for building zone thermal responses, Energy Build. 183 (2019) 538–546.

[16] G.E. Box, D.R. Cox, An analysis of transformations, J. Roy. Stat. Soc. B 26 (2) (1964) 211–243.

[17] J. Osborne, Improving your data transformations: applying the Box-Cox transformation, Practical Assess. Res. Eval. 15 (1) (2010) 12.

[18] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O’Reilly Media, 2019.

[19] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. 2 (1–3) (1987) 37–52.

[20] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Ann. Data Sci. 2 (2) (2015) 165–193.

[21] L. Ertoz, M. Steinbach, V. Kumar, A new shared nearest neighbor clustering algorithm and its applications, in: Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining, vols. 105–115, 2002.

[22] A. Espín, in: IMPLEMENTATION and EVALUATION of the SHARED NEAREST NEIGHBOR CLUSTERING ALGORITHM, Tech. Rep, Universitat Politècnica de Catalunya, 2019.

[23] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 790–799.

[24] T. Van Craenendonck, H. Blockeel, Using Internal Validity Measures to Compare Clustering Algorithms, in: Benelearn 2015 Poster presentations, 2015, pp. 1–8 (online).

[25] F.S. Westphal, R. Lamberts, Building simulation calibration using sensitivity analysis, in: Ninth International IBPSA Conference, vols. 1331–1338, 2005.

[26] I. Danielski, M. Fröling, A. Joelsson, The impact of the shape factor on final energy demand in residential buildings in nordic climates, in: World Renewable Energy Forum, WREF 2012, Including World Renewable Energy Congress XII and Colorado Renewable Energy Society (CRES) Annual Conference, 2012, pp. 4260–4264. Denver, CO; 13 May 2012through17 May 2012; Code94564.

[27] S. Hyde, Likelihood Based Inference on the Box-Cox Family of Transformations: SAS and MATLAB Programs, Dept of Mathematical Sciences, Montana State University, Billings 34p .

[28] M.F. Ijaz, G. Alfian, M. Syafrudin, J. Rhee, Hybrid prediction model for type 2 diabetes and hypertension using dbscan-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest, Appl. Sci. 8 (8) (2018) 1325.

[29] M.N. Gaonkar, K. Sawant, AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset, Int. J. Adv. Comput. Theory Eng. 2 (2) (2013) 11–16.

[30] T.M. Kodinariya, P.R. Makwana, Review on determining number of cluster in K-means clustering, Int. J. 1 (6) (2013) 90–95.