Energy & Buildings 224 (2020) 110228

Contents lists available at ScienceDirect

Energy & Buildings

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

Unsupervised learning for feature projection: Extracting patterns from multidimensional building measurements

Chunze Xiao[a], Fazel Khayatian[b,*], Giuliano Dall'O'[c]

[a] The Bartlett School of Environment, Energy and Resources, University College London, London, United Kingdom
[b] Urban Energy Systems Laboratory, Swiss Federal Laboratories for Materials Science and Technology, Empa, Dübendorf, Switzerland
[c] Department of Architecture, Built Environment and Construction Engineering, Polytechnic of Milan, Milan, Italy

ARTICLE INFO

Article history:
Received 12 February 2020
Revised 10 May 2020
Accepted 6 June 2020
Available online 11 June 2020

Keywords:
Unsupervised learning
Building performance
Dimensionality reduction
Data representation

ABSTRACT

Data visualization is an important resource for decision makers to obtain information from large datasets. Based on the data obtained from either predictions or measurements, different strategies are combined and tested to reduce the energy demand, whilst keeping the indoor comfort at suitable level. Although the information expressed from data representation can significantly influence the decisions, little research has focused on extracting features from building measurements. This paper provides an in-depth view into representation of building data, and applies three dimensionality reduction algorithms Principle Component Analysis (PCA), autoencoder and t-Distributed Stochastic Neighbour Embedding (t-SNE) on measurements from a teaching building. Results show that whilst PCA returns linear representations, it also has the least data compression, which can be useful for obtaining more general features. On the other hand, t-SNE returns the most compressed data, which is suitable for seeking large margins within a dataset. However, t-SNE may be unsuitable for datasets with recurring step-like temporal profiles. Autoencoder is the best overall option, as they capture the nonlinearities within a dataset whilst avoiding excessive data compression. Fine-tuning the hyperparameters of studied the algorithms, and the perils of relying on poorly tuned models is discussed at the end of the study.

1. Introduction

Nowadays, climate change and the shortage of resources are amongst the main concerns of the scientific community. The building sector, which is a major contributor to the problem, therefore, cannot be ignored. According to IEA (2019), the construction sector is responsible for 36% of the global energy use and roughly 40% of CO_2 emissions [1]. Moreover, large amounts of non-CO2 greenhouse gases such as halocarbons, Chlorofluorocarbons (CFCS), hydrochlorofluorocarbons (HCFCS) and hydrofluorocarbons (HFCs) is released from buildings and its construction for daily use [2]. Various building energy saving codes are published by different countries to reduce the building energy consumption, (e.g. ASHARE 90.1 and IECC in US [3], CIBSE Guide F in UK, and Energy Performance of Buildings Directive in EU [4]). However, in 2016, 30% of the global energy use and 28% of the global energy-related CO_2 emission were attributed to the building sector [5]. Therefore, great effort for reducing the energy consumption of buildings is essential for the next decades.

1.1. Data visualization for decision making

Since occupants spend 90% of their time indoors [6], alongside population growth, the continuous increase in building energy use is primarily associated with satisfying people's demands such as thermal and visual comfort, or other energy-intensive services for information, communication and entertainment [7]. Therefore, to succeed in reducing building energy use in the public sector, it is equally important to consider people's requirements for indoor comfort. This balance is highly dependent on assumptions about a building's performance, be it energy predictions obtained through simulations, or data acquisition by monitoring and measurements. In either case, during the process of analysing a building's performance, information loss is inevitable. This challenge is specifically important when treating multivariate datasets, in which information loss may distort the perception of the actual performance. According to a survey, 34% of the total responses from users indicated levels of dissatisfaction with the representation of the results from simulations [8]. Meanwhile, accurate and

[*] Corresponding author at: BA 303, Uberlandstrasse 129, 8600 Dubendorf, Switzerland.

E-mail addresses: chunze.xiao.19@ucl.ac.uk (C. Xiao), fazel.khayatian@empa.ch (F. Khayatian), giuliano.dallo@polimi.it (G. Dall'O').

https://doi.org/10.1016/j.enbuild.2020.110228
0378-7788/© 2020 The Authors. Published by Elsevier B.V.

understandable data is the basis of reasonable decision-making. For example, data from building energy prediction could help to check the effectiveness and efficiency of different designs, operation schedules, as well as managing the demand and supply relationship [9]. On the other hand, improper decisions made from poor quality data could have adverse consequences in economic and social aspects, including lower performance, higher running and maintenance cost, and dissatisfaction of customers [10], all of which, lead to additional efforts for compensation. In 2016, 3.1 trillion dollars was spent in the United States as result of poor data quality [11]. Currently, there are many studies focused on generating more accurate prediction models on building energy consumption, however, few are dedicated to the suitability of representation and visualization. Therefore, it is vital to discuss how data representation influences the perception of the information we obtain from raw data, and how to avoid information loss when summarizing information from large datasets.

1.2. Literature review

1.2.1. Building performance representation in the literature

The most common method to display building performance is line plot, which is useful for presenting trends of time-dependant variables such as energy use [12], electricity consumption [13], as well as temperature and humidity [14]. Although these plots may include multiple lines for comparison purposes as well as colour division on the chart background to contrast different categories [15], the correlation amongst different lines (i.e. variables) may be difficult to comprehend. This is particularly challenging when plotting more than two dimensions, as matching units can be cumbersome. Although not commonly used, other figure types such as area chart and pie chart may also be useful to contrast a value against the share of different segments [16,17]. Scatter plot is another commonly used data presentation technique, which is particularly useful for analysing covariations between two variables. For instance, contrasting power demand against outdoor temperature, solar radiation, or humidity [18]; as well as plotting data-driven energy predictions versus white-box simulations [19]. It is important to note that scatter plots do not preserve the internal consistency of a variable, such as data sequences, and therefore, are not particularly suitable for displaying trends of time-dependant variables. Stacked histograms are also widely used for comparison purposes; yet, they are most useful for studying frequencies of events [20], such as energy use [21], or performance error [22]. Similar to histograms, distributions show the probability i.e. the likelihood of one event, [23], but could also be rearranged to provide the cumulative probability [24], which is especially useful for reliability and risk analyses. Box plots are similar to distributions, but also provide coarse numerical measures of the distribution through percentiles [25]. Since each type of representation is suitable for a specific application, different plots are often mix-used to explain a dataset from different aspects. Among various plot types, line plots, area plots and pie charts are the typical numerical data representing methods. Histogram and distribution plots are regarded as statistical methods as they provide information on the dataset as a whole, rather than each individual value. Since comparing the shape of histograms and distributions in a single plot may be difficult, boxplots facilitate understanding a single plot of multiple distributions, while providing numerical details on the overall characteristics of each distribution.

1.2.2. Representing high-dimensional data

In recent years, various methods have been introduced to help with representing and processing high-dimensional data. Pilgrim et al. [26] introduced the representation of plotting data in retinal properties such as colour scale to show quantitative data such as time duration and temperature. The study used colour scale to show the number of hours (the frequency) of particular weather conditions, from which specific characteristics could be explained (e.g. data trend, patterns and outliers). It also used colour scale and size to show temperature distribution within a room, as well as carpet plot to contrast hourly and mean daily air temperatures. The aforementioned plots are practical approaches that tend to associate additional information with a single figure, however, limitations and drawbacks may still exist. For example, temperature plot on floor grid could become confusing if additional variables (e.g. humidity) are involved. By now, carpet plot is well accepted by researchers as a visualization technique to detect recurring patterns and abnormal behaviours [27–29].

Dimensionality reduction is another important technique for processing and representing high dimensional datasets. Miller, Nagy and Schlueter [30] proposed a new figure type, which efficiently shows the process of clustering and compressing data. By defining energy load range in the code and sorting them by time-intervals, suffix tree distinguished categories of motif and discord patterns of heatmaps. Such approach could be helpful in finding particular characteristic of the performance, as well as unusual behaviours. As the authors conclude in their paper, additional work is required for analysis of overlapping strange behaviours, extraction of multiple data streams, as well as more detailed parameters in smaller time intervals. All of the mentioned data processing schemes showed successful examples of data representation. However, few explicitly focus on representing high dimensional datasets, and none properly contrast results from different dimensionality reduction techniques.

1.3. Research gap

There is a limited variety of figures that are used for detailed data representation of building performance. Most studies render a dataset in several fundamental plot types without preprocessing, for instance, using line plots to show energy use in a time-series manner [31]. Moreover, there is not much comparison and combination between different types of plots, which might hinder the depth of data analysis and its rigour.

It can be inferred that information is commonly represented through 2D plots due to convenience. However, to discover the information hidden in a dataset, combining and contrasting different charts is recommended for a deeper analysis. To maximize the depth of data analysis, inclusion of many more features may become necessary, eventually leading to the "curse of dimensionality" [32], namely, "multidimensional variational problems" that hinder both the visualization and tabulation of variables. Recently, more clarity was added to the definition, as Donoho [33] called it the obvious difficulties of organized searching in high-dimensional space. Moreover, the problem of non-correspondence between plots and actual variables may become more evident when numerous variables are involved.

1.4. Justification of research and the main contributions

The conventional methods for representing building performance are subject to information loss or misinterpretation of the actual performance. Nevertheless, researchers in other fields especially statistics and artificial intelligence have intorduced various methods of data processing in recent years. Although very few research have focused on preprocessing and representation of building performance, researchers frequently resort to statistical analysis and machine learning, in their studies. These statistical and machine learning tools also have high potential in data representation, three of which are compared in this study.

The aim of this study is to introduce a framework for representing high-dimensional datasets of building performance by means of dimensionality reduction. This will directly benefit the designer or decision-maker who seeks beyond the shallow surface of information and looks for hidden structures within the dataset.

The main contributions of this study are:

- Providing a scope of data representation in building performance literature.
- Contrasting three of the most popular algorithms of dimensionality reduction on a case-study teaching building in Milan, Italy.
- Providing insights on the applicability of data representation by using machine learning techniques.
- Underlining the robustness to multimodality and sensitivity to hyperparameters, as well as recommendations for fine-tuning autoencoder and t-SNE.

The remainder of the paper is as follows: Section 2 provides a theoretical background on the dimensionality reduction algorithms. Section 3 introduces the case study, describes the data and compares results from the three models. Section 4 draws the conclusions and provides recommendations for future research.

2. Methods

In this section, three different dimensionality reduction techniques are introduced. The following techniques have been frequently used for visualization of high dimensional datasets. We also distinguish between linear and nonlinear dimensionality reduction techniques, whilst comparing their flexibility and ease of use.

2.1. Linear feature learning

2.1.1. Principle Component Analysis (PCA)

PCA is a typical dimensionality reduction technique for finding correlations in a dataset. The principal components represent linear hidden structures within the original dataset. The first principal component explains the largest inter-correlation within the dataset. Therefore, using the main principal components can reduce a dataset with high dimensions and massive inter-correlated variables to a few features with primary characteristics. This will help avoiding hypothetical and uncorrelated components (e.g. noise), which could interfere with the extraction of information [34].

The matrix of the principal components is:

$$z = W^{T} x^{(i)} \tag{1}$$

where, W is the transpose of the feature vector, which could be obtained by using Singular Value Decomposition (SVD), and $x^{(i)}$ is the original dataset. Once a PCA is trained, the learned matrix of coefficients (W^{T}) can be later used for extracting principal components from other datasets with the same number of features.

2.2. Nonlinear feature learning

2.2.1. Autoencoder

Sparse autoencoder is an unsupervised learning algorithm for dimensionality reduction and is based on feedforward neural networks. The purpose of using a sparse autoencoder is to learn a representation after compressing the dimensions, and then reconstructing the original data with minimum reconstruction error [35]. Autoencoders are trained by passing the inputs through two sets of hidden layers, commonly known as encoder and decoder. Similar to a multilayer perceptron, autoencoder's activation functions can be linear or logistic. Given that this section seeks manifold learning capabilities, the following descriptions will only focus on autoencoders with logistic activation functions. Encoder and decoder hidden layers may consist of similar or dissimilar activation functions, depending on the objective of the dimensionality reduction. The process of dimensionality reduction with sparse autoencodeing is to simply multiply the original dataset (inputs) by a "sparse" matrix, commonly known as encoding, and then reconstructing the sparsed data back to its original space using another matrix called the decoder. The Mean Squared Error (MSE) between input data and the reconstructed one defines the performance of the autoencoder, which is measured at each iteration of training. Once an autoencoder is sufficiently trained, the first hidden layer (i.e. encoder) is extracted from the model, and its weights and bias are used for encoding the high dimensional data into a lower dimension.

The loss function of the sparse autoencoder (MSEsparse) is:

$$J_{sparse}(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\|^{2}\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_i}\sum_{j=1}^{s_{i+1}}\left(W_{ji}^{(l)}\right)^{2} +$$

$$\beta\sum_{j=1}^{s_2} KL\left(\rho\|\hat{\rho}_j\right) \tag{2}$$

The first term is the average sum-of-the squared error, where x is the input and y is the reconstructed output. The second term is the regularization, where λ is the weight decay parameter. The third term is the sparsity penalty, in whichβ represents the weight of the penalty. KL is the Kullback-Leibler divergence function, which prevents the encoding matrix to converge into an identity matrix, and is defined as:

$$KL\left(\rho\|\widehat{\rho}_j\right) = \rho\log\frac{\rho}{\widehat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\widehat{\rho}_j} \tag{3}$$

where, ρ is sparsity parameter, which is typically a small value such as 0.05. $\hat{\rho}_j$ is the average activation of unit j with given input x, and is defined as:

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}\left((x)^{(i)}\right)\right] \tag{4}$$

where, $a_j^{(2)}$ is the activation of the hidden unit.

Autoencoders are often trained with a single hidden layer [36], however, they can adopt deep architectures as well. This study suffices to a single layer architecture as deep autoencoders are mostly used along with a multilayer perceptron. Although a detailed study of deep autoencoders is out of the scope of this paper, the advantages of deep architectures for unsupervised learning should not be underestimated [37], and could be a potential topic for future studies. As mentioned earlier, principal components obtained from PCA are linear features which may lead to information loss. However, autoencoders, have the ability to extract manifolds, which reduces the risk of misrepresenting nonlinearities within the dataset [38]. Similar to PCA, the transfer matrices learned from training an autoencoder can be later used to encode another set of data. This means that similar to PCA, autoencoders are ideal preprocessing tools for denoising or sparsing data prior to the main analysis.

2.2.2. t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a dimensionality reduction algorithm first introduced by L.J.P. van der Maaten and G.E. Hinton [39]. Originated from Stochastic Neighbor Embedding (SNE), it is found suitable for visualizing high dimensional datasets. t-SNE firstly creates probability distribution by converting high-dimensional Euclidean distances between data points to represent their similarity. Given data point x_j andx_i, the conditional probability $p_{j|i}$ is defined by

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2/2\sigma_i^2\right)} \tag{5}$$

where, σ_i is the standard deviation of data point x_i. The probability could be determined through binary search with a fixed perplexity, which is defined as

$$Prep(P_i) = 2^{H(P_i)} \tag{6}$$

where $H(P_i)$is the Shannon entropy of P_i defined as,

$$H(P_i) = -\sum_j p_{j|i}\log_2 p_{j|i} \tag{7}$$

and for data points pairwise purpose,j≠i.

Then, in low-dimensional counterparts y_j and y_i the similar conditional probability $q_{j|i}$ is

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k\neq i}\exp\left(-\|y_i - y_k\|^2\right)} \tag{8}$$

where, σ_i is set as $\frac{1}{\sqrt{2}}$ i.e. the variance of Gaussian distribution under low dimensional result y_i , and j≠i.

The cost function is

$$C = \sum_i KL(P_i\|Q_i) = \sum_i\sum_j P_{j|i}\log\frac{p_{j|i}}{q_{j|i}} \tag{9}$$

where, KL is Kullback-Leibler divergence, as previously described in section 2.2.1, P_i is conditional probability distribution over all other data points for given x_i and Q_i is the conditional probability distribution over all other map points for given y_j.

Unlike PCA and autoencoder, the feature extraction (dimensionality reduction) strategy of t-SNE is hardly repeatable. This is because t-SNE does not learn a function for transferring data to another dimension. Since there are no learned functions after training t-SNE, we cannot reuse the model for compressing other datasets. It also indicates that with t-SNE, it is impossible to reconstruct the original dataset from lower dimensions.

3. Case Study

In this section, the dimensionality reduction techniques previously discussed in section 2 are applied to a dataset of observations from a case-study building in Italy. The case-study is a teaching building at the Bocconi University campus, located in the southern part of Milan city. The building has four stories, of which floors 1 to 3 have the same plan with 10 classrooms (Fig. 1). The ground floor has five classrooms and the rest of its space is dedicated to the reception and lobby.

3.1. Data acquisition and preprocessing

The dataset is composed of three main types of features as displayed in Table 1. First, ***climatic*** data in the form of Actual Meteorological Year (AMY) weather file, which is collected from an onsite weather station. Second, ***operational*** data, which is collected or derived from onsite measurements of occupants and equipment. Third, onsite measurements of the building's ***performance*** i.e. energy consumption of the heat pump, as well as the supply and extract air temperatures at various classrooms.

Climatic data are measured onsite by a weather station located on the university campus. The weather station is placed on a rooftop, roughly 50 m from the case-study building, and managed by the "*Milano Duomo Meteorological Observatory Foundation*" [40]. Indoor air temperature, lighting, equipment and heating energy consumption measurements are collected from a SIEMENS Building Automation and Control system. Measurements cover more than four months from December 1st 2016 to April 5th 2017. Indoor air temperature is measured in 30-minutes intervals by SIEMENS QPM21 series sensors at both inlet and exhaust ducts of each classroom. However, this study only focuses on measurements at the classroom exhaust node, to observe the effect of occupants on indoor air temperature. Indoor air temperature is only studied for a candidate classroom on the second floor as highlighted in Fig. 1 with an asterisk. Missing values and anomalies share less than 0.2% of the indoor air temperature dataset and do not exceed five consecutive time-steps, hence, are infilled with linear interpolation. The operational profiles (occupancy, lighting, equipment) are obtained from a single classroom, and considered representative of the entire building. Such assumption although not ideal, can be justified by the fact that the whole building is mostly composed of classrooms with identical usage schedules. Lighting and plug loads are measured in hourly intervals from the candidate classroom with no missing values or anomalies, and then resampled to 30-minutes temporal resolutions with linear interpolation. The profile of occupant density is estimated from the lecture schedules of the candidate classroom, as well as the number of students in the registry list of each lecture. The occupant density profile is also compiled in 30-minutes intervals. The heating energy consumption of the heatpump has a 10-minute temporal resolution with no outliers or missing values, and is resampled to 30-minute intervals with linear interpolation.

The complied dataset (Table 1) is used as a source to examine the applicability and practicality of the proposed dimensionality reduction techniques. The ***operational*** and ***climatic*** set of features are concatenated as a single input for dimensionality reduction. The ***performance*** features i.e. indoor temperature and energy consumption are chosen as metrics for assessing the lower dimension features. Since the heating system was turned off from March 17 onwards, the heating energy consumption dataset is shorter than that of indoor air temperature.

3.2. Manual extraction of patterns

In this section, we demonstrate how dimensionality reduction can be useful for manual identification of patterns within a dataset. Unlike machine-learnt clustering which is fully unsupervised (i.e. has no target), manual partitioning of a dataset enables us to extract patterns which co-vary with an intended target. In this study, the intended targets for extracting patterns is either the indoor temperature, or the heating energy consumption.

Different dimensionality reduction methods have dissimilar responses to the range of data. Therefore, all features (operational, climatic, and performance) are rescaled between zero and one.

There are different ways to stop the training process of unsupervised learning, e.g. tolerance of the error, convergence of the

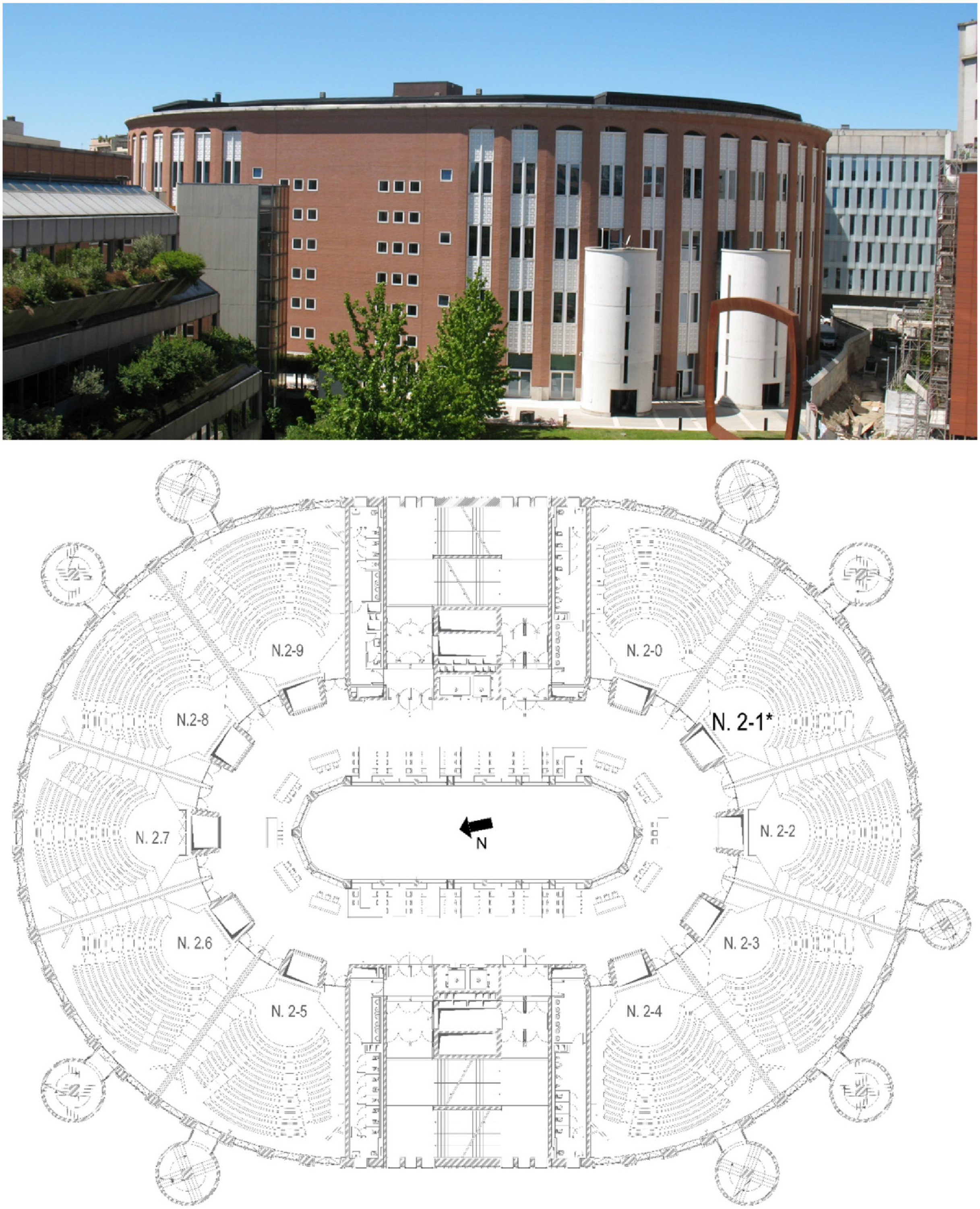

Fig. 1. The case study building (top), and its typical floor plan (bottom).

trained weights, or the number of training iterations. Nevertheless, some thresholds can be created on an ad hoc basis, and therefore, may be tailored to a specific training algorithm. Moreover, the magnitude of error in different algorithms can greatly vary, even if the same metric (e.g. MSE) is used to evaluate the performance. Therefore, in this study we use the maximum number of iterations as the stopping criteria for all three dimensionality reduction algorithms. To ensure that all three algorithm have converged, we compared their performance at 100, 1000 and 10,000 iterations as the stopping criteria. It was observed that for our specific dataset, the improvement of the MSE after 1000 iterations is negligible.

Each training algorithm also requires some fine-tuning, namely, adjusting the hyperparameters to obtain the best performance for a specific dataset. If a dataset has sufficient samples and no missing values, using SVD returns the best accuracy in PCA. Aside from choosing the training algorithm, PCA does not require any specific

Table 1
Input features and target metrics.

	Feature Type	Feature Name	Source of Data
9d input features	Operational	Occupant Density	Calculated from the number of students registered at each class
	Operational	Equipment power Density	Onsite measurements
	Operational	Lighting power density	Onsite measurements
	Climatic	Solar gain	Calculated from onsite measurements of global solar radiation
	Climatic	Outdoor dry bulb air temperature	Onsite measurements
	Climatic	Outdoor wet bulb air temperature	Calculated from onsite measurements of dry bulb temperature and relative humidity
	Climatic	atmospheric pressure	Onsite measurements
	Climatic	wind speed	Onsite measurements
	Climatic	wind direction	Onsite measurements
target metrics	Performance	Indoor air temperature	Onsite measurements: VAV system's outlet in one classroom
	Performance	Heating energy consumption	Onsite measurement

fine-tuning in MATLAB. On the other hand, autoencoder has many characteristics that should be adjusted for optimal performance, among which, the transfer functions and the regularization weights are most important. In MATLAB, two types of logistics functions are available for autoencoders, i.e. logistic sigmoid and saturating linear functions. Although both are amongst the most popular activation functions, saturating linear functions suffer from two issues that are particularly important in shallow neural networks (as used in this study). First, they are non-differentiable at zero, and second, they are unbounded (no upper limit). We tried both functions on our database and observed that logistic sigmoid produces better results than saturating linear fucntion, regardless of the number of neurons. It should be noted that MATLAB does not natively support advanced optimization algorithms (e.g. adam) for training shallow networks, which might have affected the choice of the logistic function in this study. Concerning autoencoder's hyperparameters, a grid search (100 samples) was conducted to find the best values of "*sparsity regularization*" and "*sparsity proportion*". Similar to autoencoder, t-SNE has two important hyperparameters that affect its performance. In fact, the sensitivity of t-SNE to tuning hyperparameters is much greater than autoencoder (see section 3.4). Therefore, a grid search (100 samples) was also conducted to seek the optimal values of t-SNE's "*perplexity*" and "*exaggeration*". Further details on the grid search and recommendations for training an autoencoders and t-SNEs are provided in section 3.4.

3.2.1. Extraction of one feature

To facilitate comparison with *target metrics*, we compress the input data (hereon called the "*9d input features*"), from nine dimensions (6 climatic + 3 operational) to one dimension (hereafter called the "*1d hidden feature*"). This dimensionality reduction is performed by three different methods as argued in section 2, i.e. PCA, autoencoder, and t-SNE. Concerning PCA, the *1d hidden feature* is obtained by extracting the first principal component from the dataset. In autoencoder, the number of neurons of the hidden layer is set to one, which returns the intended *1d hidden feature*. Autoencoder's "*sparsity regularization*" and "*sparsity proportion*" values are set to 3.49 and 0.61, respectively. The output dimension of t-SNE is also set to one, which yields the anticipated *1d hidden feature*. The "perplexity" and "exaggeration" values of t-SNE are set to 5715 and 34, respectively.

As argued before, one purpose of dimensionality reduction is learning hidden patterns with ease from multidimensional datasets. A unitless single-dimension output from dimensionality reduction can be difficult to interpret on its own, yet, it can be post-processed and cross-matched with performance characteristics. In other words, we seek to find correlations between the newly extracted *1d hidden feature*, and the building's *target metrics* (Table 1).

In this section, scatter plots are used for identifying patterns within the dataset, i.e., *1d hidden features* vs indoor air temperature (Fig. 2), as well as *1d hidden features* vs heating energy consumption (Fig. 3). In these figures, the vertical axis is the target metric, which can be either the indoor air temperature or the heating energy consumption. The horizontal axis on the other hand is a unitless vector, which encapsulates covariations within the *9d input features*. In fact, the horizontal axis is a representation of all climatic and operational variabilities compressed into a single array, which allows us to contrast a representation of all *9d input features* against the intended target metric. By visually assessing the scatter plots, groups of data are identified and then highlighted on the graphs. The chosen groups are cross-matched with each of the building's operational and climatic characteristics (*9d input features*), and the coefficient of determination (R-squared, or R^2) is used for quantifying possible covariations. The groups of points that return an R-squared value of 0.5 or higher are shortlisted amongst all visible scatter points. The shortlisted groups are then labelled with numbers on the scatter plots, to facilitate reading patterns within the dataset.

Table 2, together with Figs. 2 and 3 show how the *1d hidden features* obtained from PCA, autoencoder, and t-SNE, create distinguishable patterns when contrasted against either indoor air temperature of a classroom or the heating energy consumption of the entire building. It is important to note that the R-squared values reported in Table 2 do not indicate the covariation between building features (e.g. equipment power) and building performance (e.g. indoor air temperature). Instead, Table 2 explains how various building features (e.g. equipment power) correlate with the *1d hidden feature*. These R-squared values vary for different sections of the plot. In some instances, the *1d hidden feature* represents equipment power (Fig. 2, box 1), whilst in other cases, it represents solar gain (Fig. 2, box 3). This helps to distinguish how the *1d hidden feature* encapsulates covariations within the *9d input features* during different periods of measurement.

3.2.1.1. Correlations with indoor air temperature.

All three methods highlight solar gain and equipment power as distinguishable patterns. Autoencoder and t-SNE both find patterns that correlate with outdoor dry bulb temperature. It can be observed that PCA and autoencoder patterns have generally higher R-squared values. However, t-SNE surpasses the other two methods in term of the diversity of patterns, yet, at the cost of returning smaller R-squared values. This indicates that PCA and autoencoder find more distinguishable patterns within the dataset, whilst t-SNE extracts more diverse patterns. Autoencoder shares the most features with either PCA or t-SNE. The effectiveness of visualization varies with each method. For instance, t-SNE generally groups the data in a more distinguishable manner, when compared to the other two methods. However, such compression of data comes at the cost

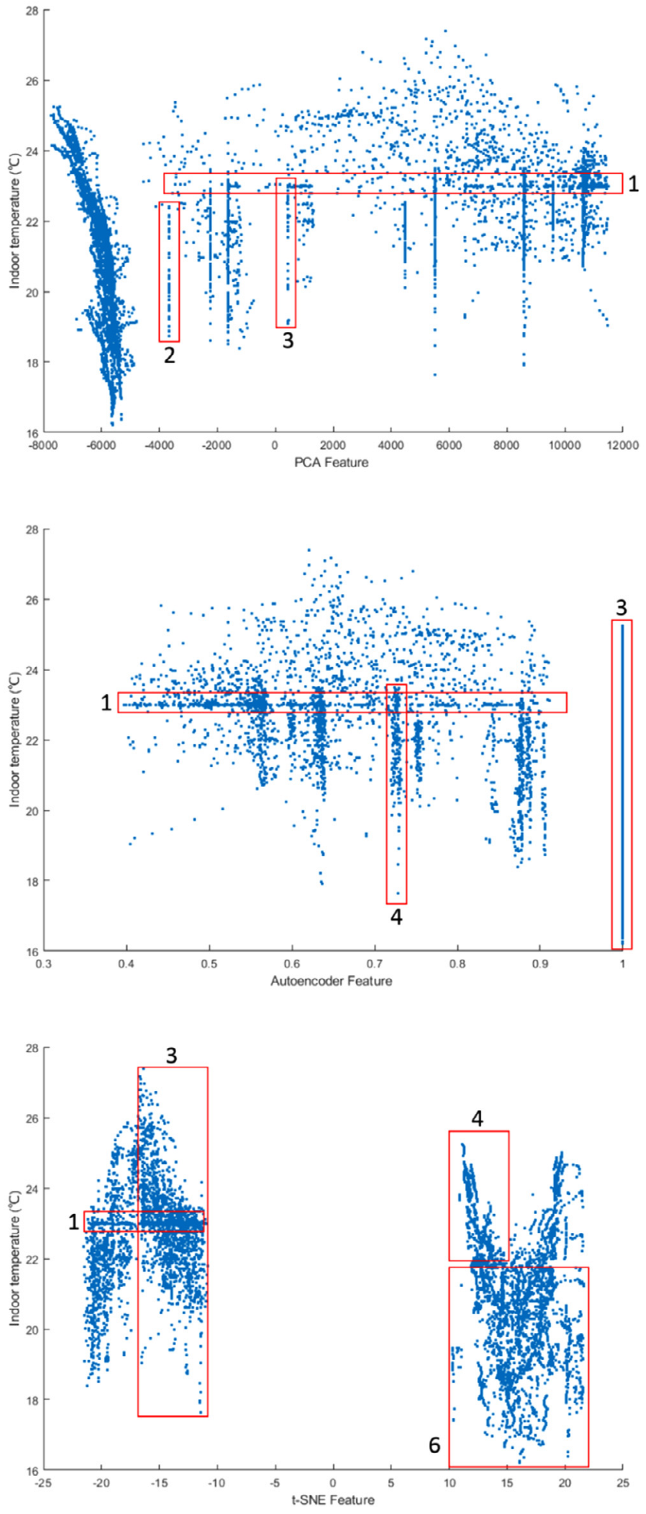

Fig. 2. Contrasting *1d hidden feature* against indoor air temperature. A comparison between PCA (top), autoencoder (middle) and t-SNE (bottom). This figure should be read together with Table 2.

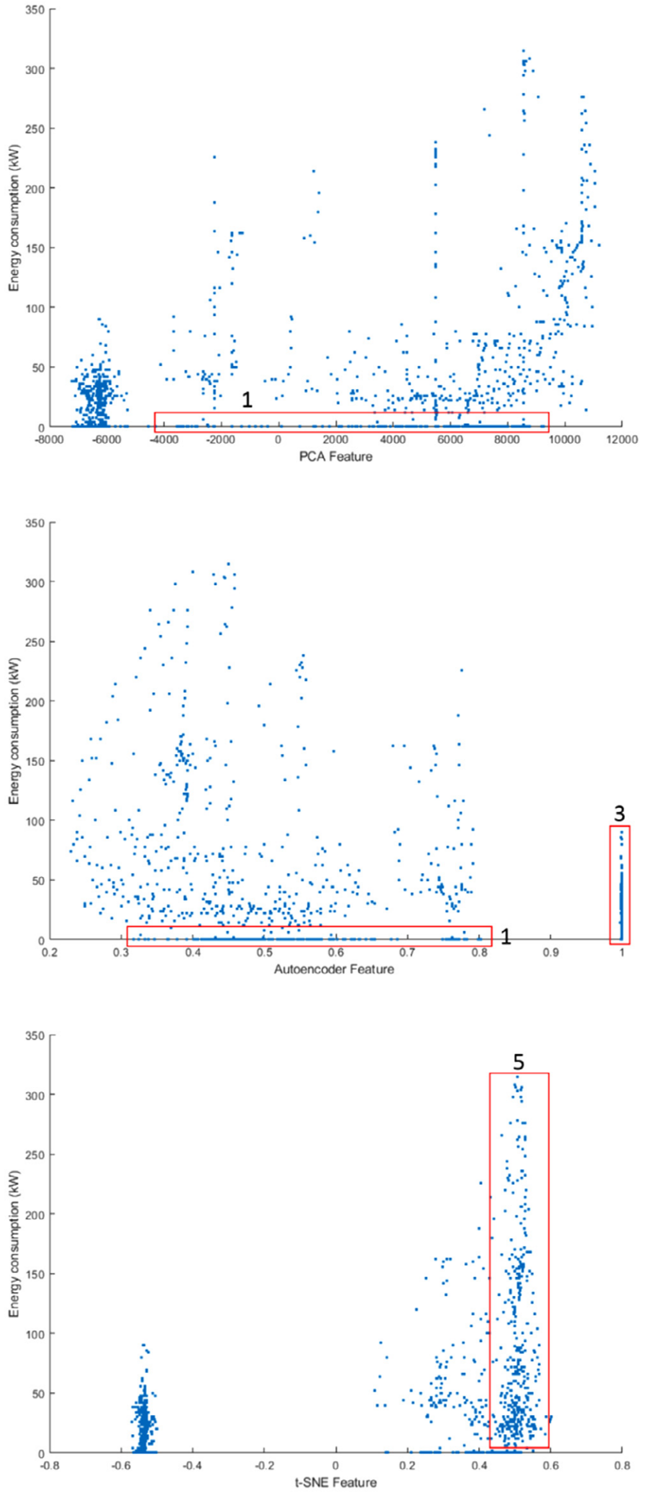

Fig. 3. Contrasting *1d hidden feature* against energy consumption. A comparison between PCA (top), autoencoder (middle) and t-SNE (bottom). This figure should be read together with Table 2.

of dense scatter clouds, within which, extracting additional patterns is almost impossible.

3.2.1.2. Correlations with energy consumption. Seeking correlations between energy consumption and the *1d hidden features* reveal few recognizable patterns. It is important to remind that although the composition of the *9d input features* are similar in Figs. 2 and 3, the number of observation differ (see section 3.1). As a result, the number of samples plotted against indoor air temperature (Fig. 2), differ from the number of samples plotted against energy consumption (Fig. 3). Autoencoder highlights patterns that co-vary with equipment power, which is also confirmed by findings from PCA's first principal component. Autoencoder also finds scat-

Table 2
Coefficient of determination (R-squared values) extracted from *1d hidden features*. This Table should be read together with Figs. 2 and 3.

	Indoor air temperature			Energy consumption		
	PCA	*AutoEnc*	*t-SNE*	*PCA*	*AutoEnc*	*t-SNE*
Equipment (1)	0.99	0.96	0.76	0.99	0.98	–
Atmospheric Pressure (2)	0.97	–	–	–	–	–
Solar gain (3)	0.87	0.91	0.65	–	0.98	–
Outdoor air temperature (4)	–	0.57	0.57	–	–	–
Occupants (5)	–	–	–	–	–	0.83
Wind Direction (6)	–	–	0.72	–	–	–

ter points that represent solar gain. On the other hand, t-SNE finds traces of occupant density. In general, few patterns are observed from heating energy consumption plots and most scatter clouds are not visually recognizable. Such poor performance in extracting patterns can be attributed to the delayed response of the heat pump to the *9d input features*, be it operational or climatic variables.

3.2.1.3. Observations from single feature extraction. The scatter plots of *1d hidden features* versus indoor air temperature are more spread and less concentrated when compared to that versus heating energy consumption. This can be associated with several factors: (1) the size of energy consumption dataset is smaller than that of indoor air temperature, (2) indoor air temperature has a faster and more direct response to the *9d input features*, (3) operational features (in Table 1) and solar gain are observed for a single classroom and generalized to the entire building, and (4) the energy consumption is highly correlated with the heat pump's operational state, which is not observed in this study.

3.2.2. Extraction of two features

A similar attempt is made to obtain two features from PCA, autoencoder and t-SNE, hereafter called the *2d hidden features*. However, in this section we will suffice to analysing correlations of hidden features with indoor air temperature. Since two hidden features are intended, the first two principal components of PCA are extracted. Autoencoder's hidden layer size (neurons) is set to two, and optimal values of "*sparsity proportions*" and "*sparsity regularization*" are set to 0.77 and 0.57, respectively. Similarly, t-SNE's output dimension is set to two, whist the "*perplexity*" and "*exaggeration*" values are defined as 6009 and 69, respectively. To facilitate reading the scatter plots, the *2d hidden features* are plotted along the horizontal and vertical axes, whilst the *target metric* is rendered over the scattered points, resembling a heat map. Similar to the previous plots, the horizontal axis is a compressed representation of the *9d input features*; however this time, the vertical axis is also another compressed representation of the *9d input features*. Therefore, both horizontal and vertical axes (Feature 1 and Feature 2) represent variations within the *9d input features*. The similarity or difference of Feature 1 and Feature 2 can highlight the power of a dimensionality reduction algorithm in compressing data, whilst preserving the inter-correlations and overall variations.

3.2.2.1. Correlations with indoor air temperature. Covariations between *2d hidden features* and indoor air temperature is presented in Fig. 4. All three dimensionality reduction methods split the dataset into two subsets, yet, this trend is graphically more distinguishable in t-SNE and autoencoder. A numerical comparison between the three methods is provided in Table 3. To demonstrate the power of each dimensionality reduction method in separating climatic and operational features, we included all the *9d input features* in Table 3. Unlike Table 2, which reports the R-squared values for selected groups of data, Table 3 shows how the entire

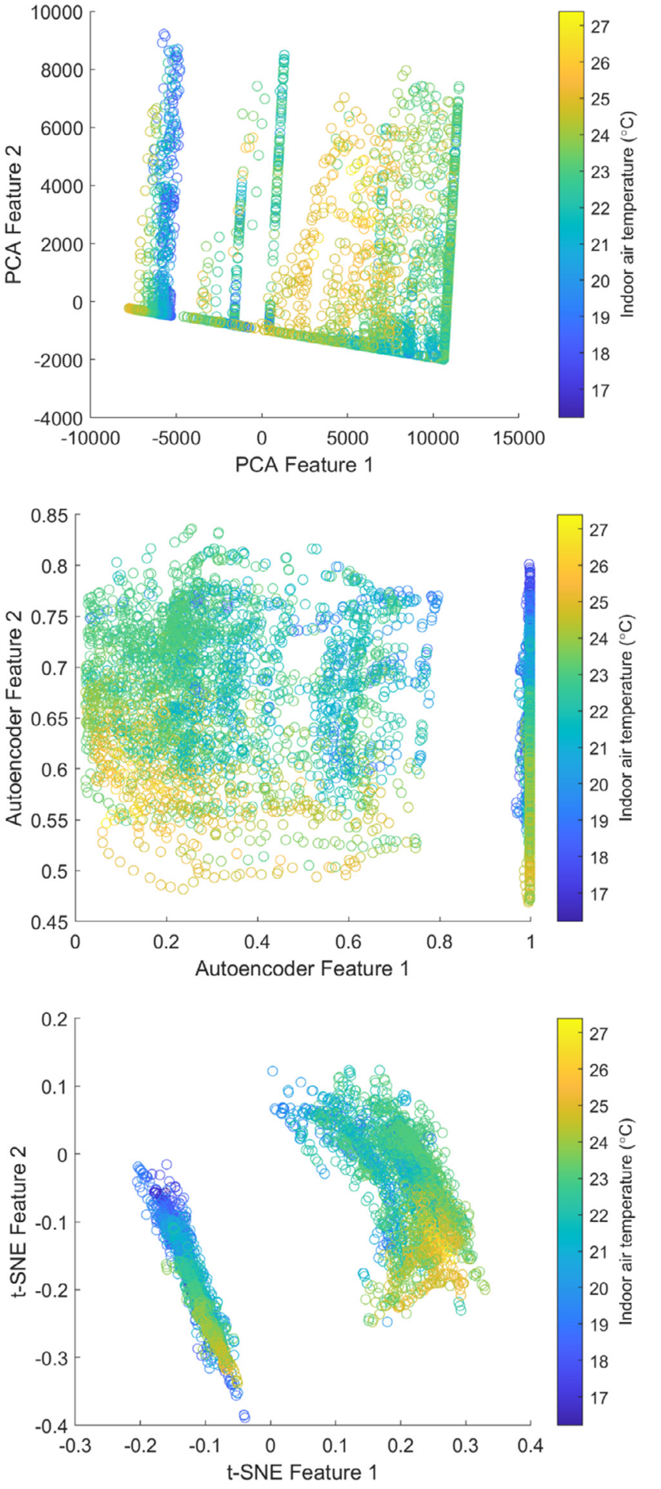

Fig. 4. Contrasting *2d hidden features* against indoor air temperature. A comparison between PCA (top), autoencoder (middle) and t-SNE (bottom).

Table 3
Decomposition of *2d hidden features*, and their covariation (R-squared) with the *9d input features*.

	PCA		Autoencoder		t-SNE	
	Feature 1	Feature 2	Feature 1	Feature 2	Feature 1	Feature 2
Occupant density	0.94	–	0.87	–	0.94	0.53
Equipment power	0.90	–	0.90	–	0.83	0.45
Lighting power	0.85	–	0.85	–	0.78	0.39
Solar gain	0.1	–	0.19	–	0.12	–
Dry-bulb temperature	–	0.72	–	0.68	–	0.22
Wet-bulb temperature	–	0.39	–	0.46	–	0.17
Atmospheric pressure	–	0.52	–	0.53	–	0.26
Wind speed	–	–	–	–	–	–
Wind direction	–	–	–	–	–	–

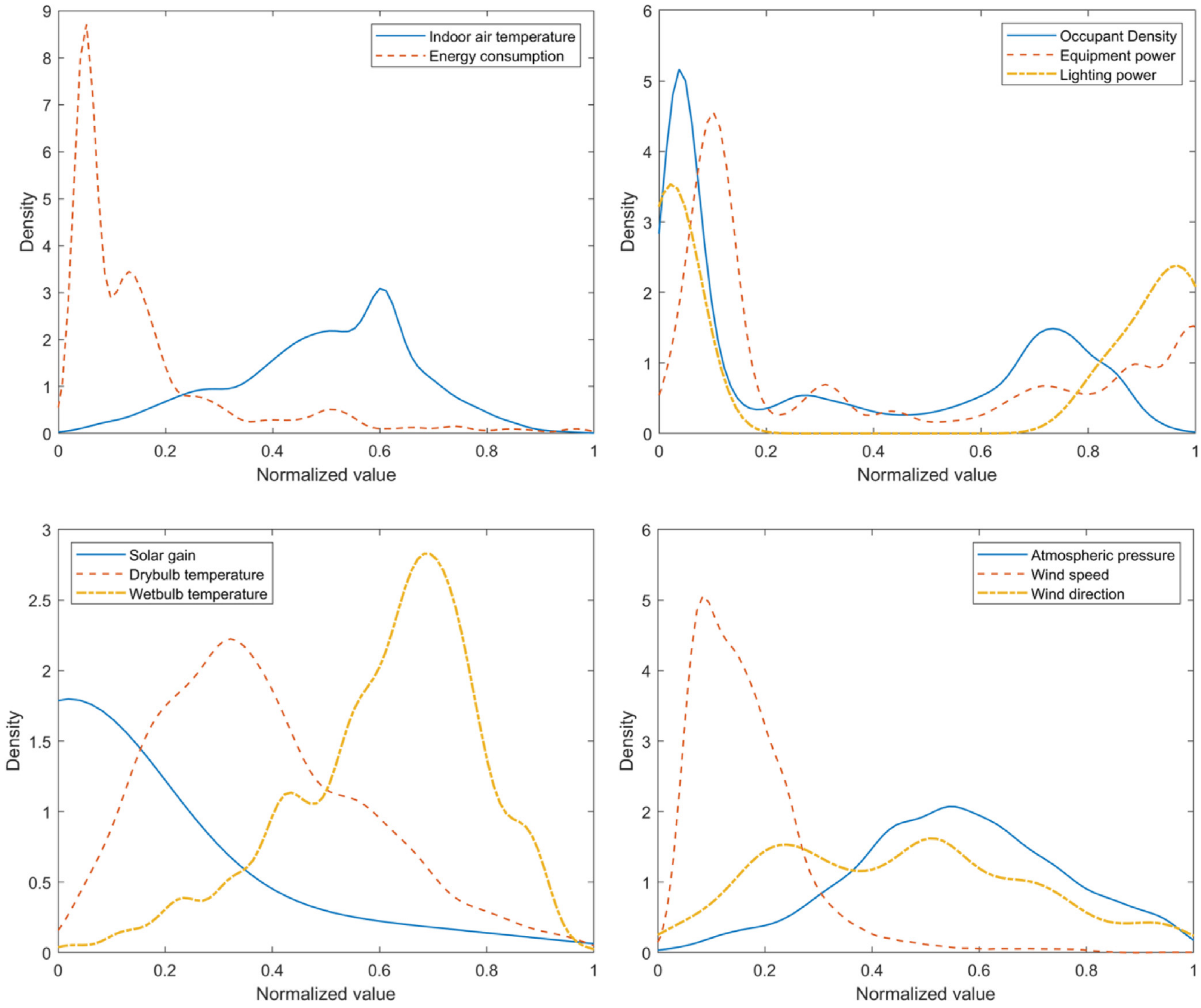

Fig. 5. Density profiles of normalized features.

hidden feature co-varies with climatic or operational features. Therefore, it is important to note that Table 2 is focused on capturing covariation within the whole dataset, i.e. how well does the *1d hidden feature* represent the *9d input features*. On the other hand, Table 3 is focused on grouping and separating the *9d input features*, i.e. how different are hidden Features 1 and 2. Results show that PCA and autoencoder draw a clear margin between operational and climatic variables. For instance in autoencoder and PCA, Feature 1 represents operational variables and Feature 2 represents climatic variables. On the other hand, both t-SNE features display covariations with operational inputs. Such observation is directly related to the shape of the distributions, and the robustness of different dimensionality reduction methods against multimodality. Interestingly, solar gain is always represented alongside operational features. This may be related to the shape of solar gain's distribution (Fig. 5). In fact, the highly skewed and truncated distribution of solar gain is quite similar to that of occupant density and lighting power.

PCA's Feature 1 co-varies with indoor air temperature, however, no monotonic patterns are observed along the horizontal axis. PCA returns equipment power, lighting power and occupant density, as the main factors affecting indoor air temperature. PCA's Feature 2 however, has little to no impact on temperature variation, as colour gradients remain constant along the vertical axis. This indicates that PCA's representation of atmospheric pressure, outdoor drybulb and wet bulb temperatures, is unable to explain variations in indoor air temperature.

Autoencoder also splits the dataset into two segments (Fig. 4, middle graph). One subset is highly densified along Feature 1 (horizontal axis), whist the other subset has a notable sprawl along both axes. It is important to note that unlike PCA, there is no particular sequence concerning the importance of hidden features obtained from autoencoder. This is due to the fact that autoencoder's hidden features are extracted from the activations of a neural network, which are randomly initialized without any particular order. Therefore, it is plausible that autoencoder's Feature 2 explains more covariations than Feature 1, or vice versa. It is observed that indoor air temperature generally decreases along the vertical axis (Feature 2), as well as the horizontal axis (Feature 1), although in a less distinguishable manner. Autoencoder's Feature 1 is highly correlated with occupant density, as well as lighting and equipment power (Table 3), both of which have a step-like temporal profile. In statistical terms, these profiles translate into bimodal probability density functions (Fig. 5, top right). This nonlinearity in the temporal profile can be roughly represented with autoencoder's log-sigmoid transfer function. However, the estimation's likelihood is highly dependent on the sparsity regularization (see sections 2.2.1 and 3.1.3). Incidentally, relative entropy (KL-divergence) which is used in autoencoder's cost function may not be the optimal measure for evaluating distribution spread. Therefore, highly asymmetric scatter clouds are inevitable when using KL-divergence in autoencoders. Feature 2 of the autoencoder is highly correlated with outdoor dry-bub and wet-bulb temperatures, as well as atmospheric pressure, all of which have unimodal distributions (Fig. 5, bottom right and bottom left). As a result, autoencoder's Feature 2 provides a continuous spreads along the vertical axis. This spread enables us to distinguish patterns of temperature change relative to autoencoder's hidden features.

Fig. 4 (bottom) shows the covariation of t-SNE's features with indoor air temperature. Since t-SNE doesn't preserve the shape of distributions, its hidden features do not distinguish between climatic and operational data. Furthermore, given that both Features 1 and 2 correlate with operational inputs, the two subsets are separated by a diagonal gap. Namely, the bimodal distributions of operational inputs are reflected in both Features 1 and 2. Such representation overcomplicates reading the scatter plot, as no clear distinction between operational and climatic inputs is possible. Furthermore, t-SNE's over-compression of the inputs result in dense scatter clusters, which impedes reading additional patterns within each scatter cloud.

3.2.2.2. Observations from two feature extraction. It is observed that using PCA for feature projection is suitable for distinguishing different profiles with dissimilar temporal patterns. PCA's two principal components successfully highlight the covariation between inputs and indoor air temperature. However, the linear representation of PCA results in information loss, which in the worst case results in overlooking some minor patterns. In this study, PCA's limitation was highlighted by fragmented linear patterns in the scatter plot.

Autoencoder, surpassed the other two methods in representing diversity within the two features, specifically, avoiding overfitting onto a single input. As a result, autoencoder's *2d hidden features* provide the most distinguishable patterns of covariation amongst all scatter plots. However, the KL-divergence regularization used in autoencoders are suboptimal for representing distribution spread. In this case-study, autoencoder over-compressed one subset and masked the patterns within that particular cluster.

It is believed that t-SNE's *2d hidden features* are the least suitable for manual extraction of patters from a graph. This is because t-SNE may associate an input with both hidden features, and consequently diminish the diversity of the inputs in the *2d hidden features*. For instance in this case-study, t-SNE's limitation resulted in underrepresenting unimodal distributions (climatic inputs) in the *2d hidden features*.

3.2.3. Dimensionality reduction for assessing machine-learnt clusters

In sections 3.2 we contrasted the suitability of PCA, autoencoder, and t-SNE for manual extraction of patterns from multidimensional spaces. However, dimensionality reduction can also be useful for assessing unsupervised machine-learnt partitioning i.e. clustering. Using reduced spaces can highlight the sensitivity of a clusters to specific inputs. In this study, we use k-means as a conventional clustering technique to partition the *9d input features* into a number of subsets [41]. The "silhouette" criterion is utilized for assessing the goodness of the clustering fit, i.e. maximizing intra-cluster distances, whilst minimizing that of inter-clusters [42]. Although a parametric search indicates that two subsets is the optimal configuration (Fig. 6), for completeness, we also contrast the suitability of using dimensionality reduction for visualizing three, four, and five subsets.

Fig. 7 renders two clusters over the *2d hidden features* scatter plots. It is observed that regardless of the dimensionality reduction method (be it PCA, autoencoder or t-SNE), the two hidden features can properly explain the hyperplane which separates the two clusters. Given that autoencoder and t-SNE's capture nonlinearities within the input dataset, they are able to find a hyperplane with a larger margin. It is perceived that the *2d hidden features* from all three methods can also adequately explain clustering with three subsets (Fig. 8). Once again, hidden features from t-SNE and (to a lesser extent) autoencoder draw clearer margins than PCA, yet none of the three methods can strongly distinguish between clusters 2 and 3. When clustering into four subsets, all three methods fail to distinguish between Clusters 2 and 4 (Fig. 9). Increasing the

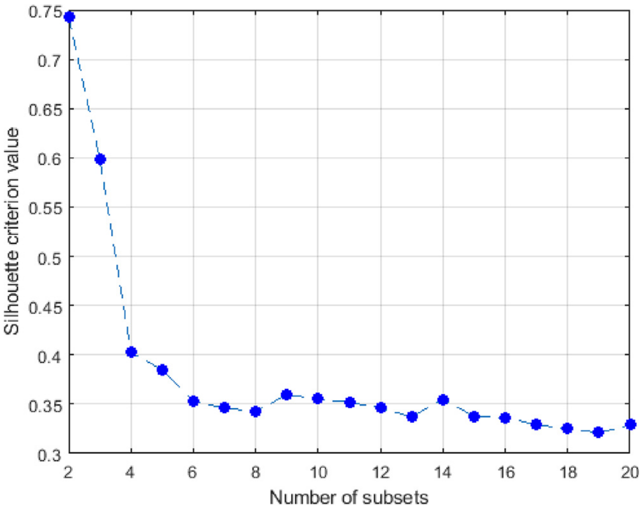

Fig. 6. Parametric search for the optimal number of subsets using Silhouette criterion. Larger Silhouette values correspond to better clustering.

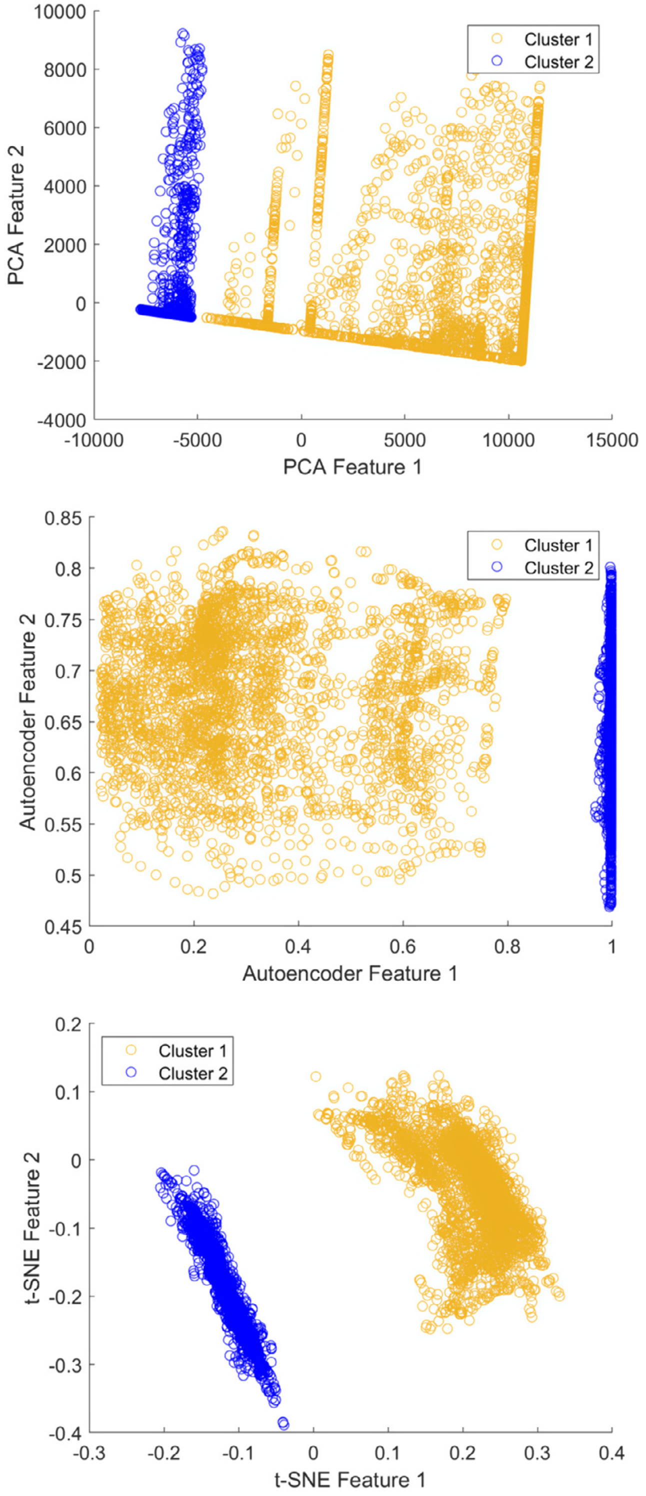

Fig. 7. Using 2d hidden features to assess clustering with two subsets. A comparison between PCA (top), autoencoder (middle), and t-SNE (bottom).

number of clustered subsets to five, reveals how a heavy compression of data can be counterproductive (Fig. 10). Whilst none of methods can distinguish between subsets 2 and 5, autoencoder and PCA display a better separation of clusters 1, 3 and 4, thanks to their highly scattered plots. In the meantime, t-SNE draws the least distinguishable margin.

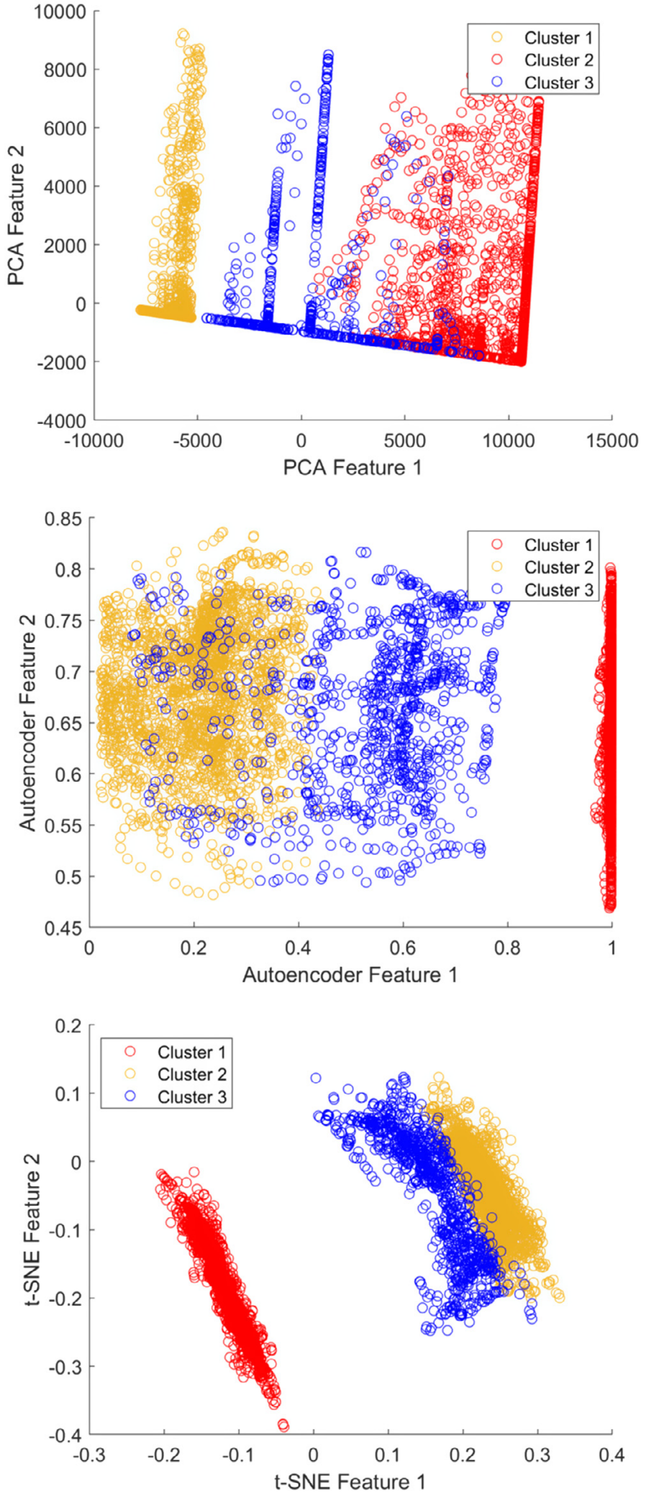

Fig. 8. Using *2d hidden features* to assess clustering with three subsets. A comparison between PCA (top), autoencoder (middle), and t-SNE (bottom).

A noticeable characteristic recurring in autoencoder scatter plots is the superior separation of clusters. This underlines the strength of autoencoders in extracting nonlinear features from a dataset, with the least amount of information loss. Although PCA does not draw the clearest hyperplanes for any number of clusters, it keeps a balance between data compression and segmentation. Contrary to PCA, t-SNE performs best when fewer clusters are intended, given its strong data compression characteristics.

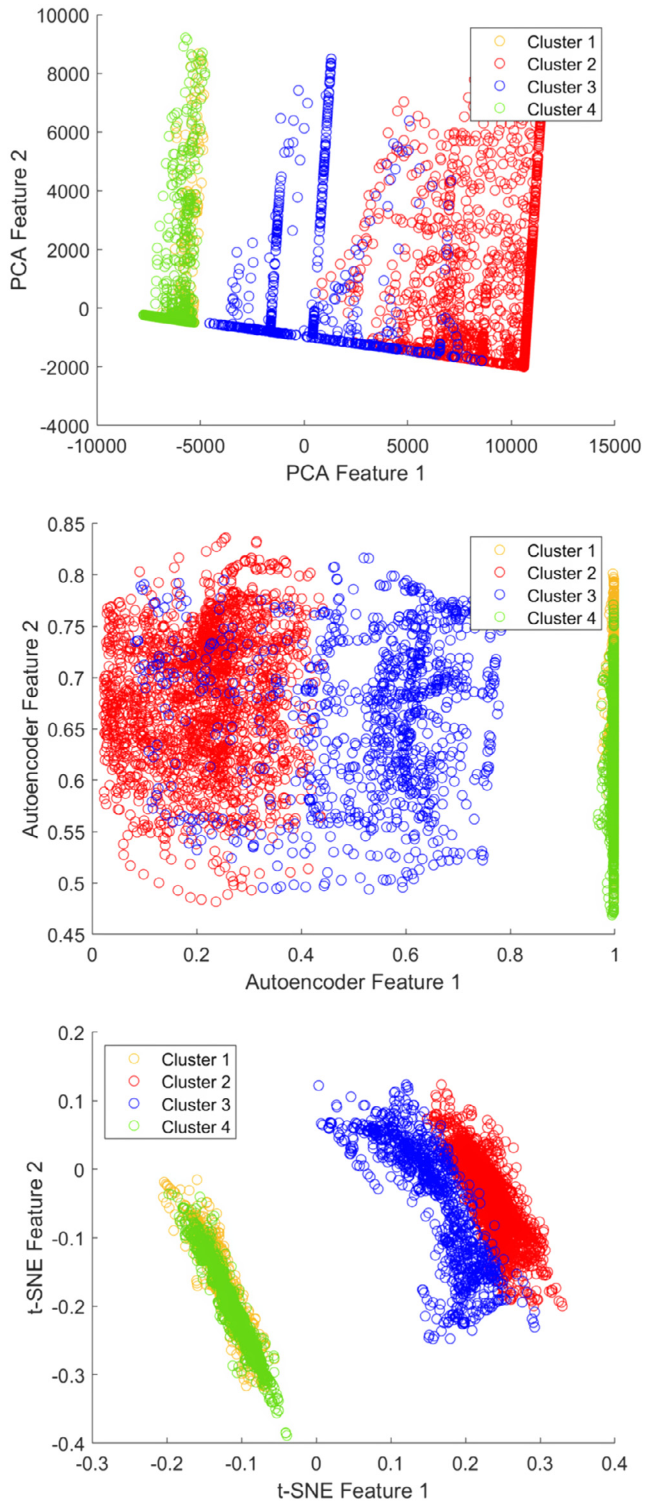

Fig. 9. Using *2d hidden features* to assess clustering with four subsets. A comparison between PCA (top), autoencoder (middle), and t-SNE (bottom).

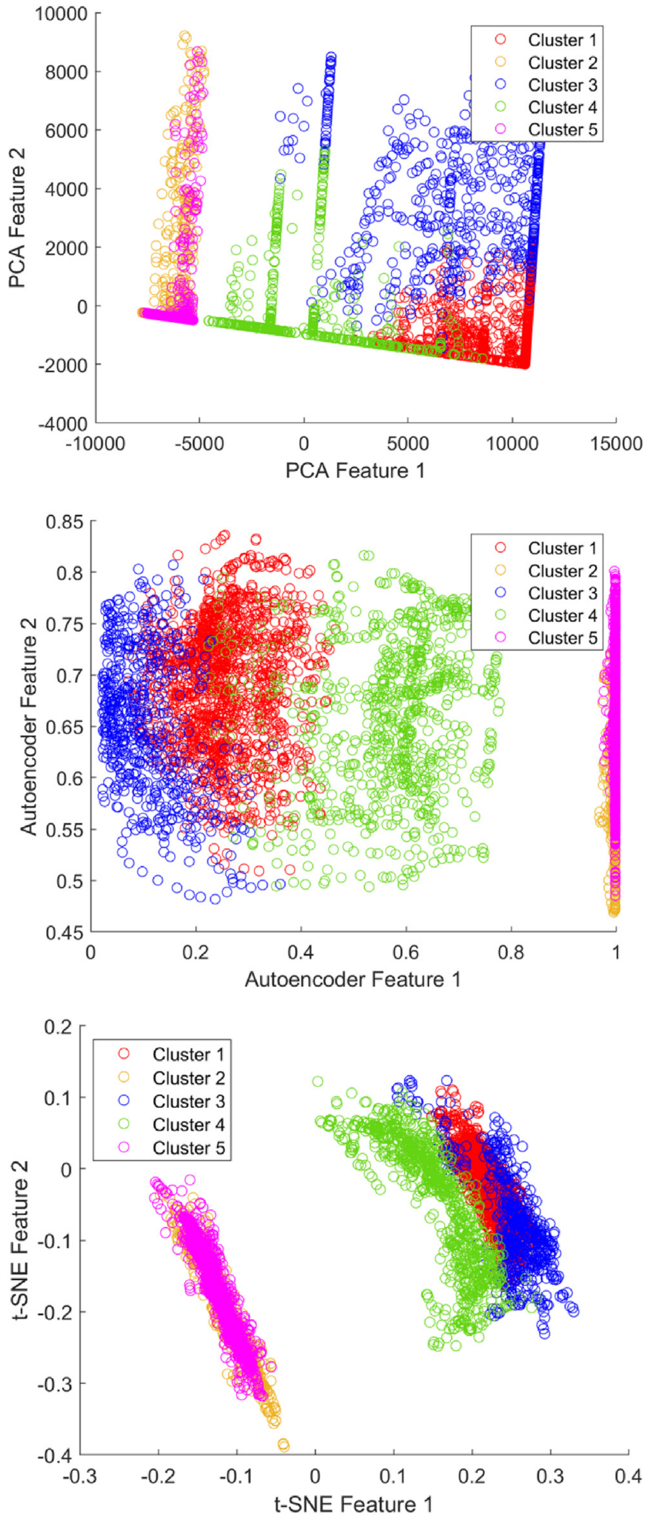

Fig. 10. Using *2d hidden features* to assess clustering with five subsets. A comparison between PCA (top), autoencoder (middle), and t-SNE (bottom).

3.3. Applications to mining building performance

A building's response to indoor and outdoor variations can be rather complex. Most studies suffice to contrasting two large sets of observations against each other, e.g. indoor air temperature versus outdoor dry-bulb temperature. There are two important issue with such analysis. First, it would be impossible to account for covariations within indoor or outdoor features themselves. For instance, some features such as occupant density, lighting power, and equipment power may follow very similar patterns (Fig. 5). Without accounting for the internal covariations within a dataset, it would be impossible to seek principal features that affect a building's performance. Second, a building's response to indoor and outdoor fluctuations vary at different times of the day, week, and year. In other words, the weighted effects of indoor and outdoor phenomena on a building's performance is not constant throughout the year.

To address this issue, we need to look at chunks of data, process their internal covariation, and then contrast them against a specific metric (e.g. energy consumption). However, there is a great challenge when working with portions of data rather than the whole dataset, i.e., how to select appropriate groups of data? Some studies manually find clusters by eyeballing the scatter plots, for instance plotting all outdoor variations versus indoor air temperature. The main drawback of such analysis is that extracting covariations is only possible one at a time. Therefore, we cannot assess the joint effects of many inputs on the building's response. Feature projection is one way to address this issue and will help in explaining joint effects on a single response. Another approach to address this issue is through (semi-)automated methods to find the most important or representative groups of data, often by clustering. Whilst clustering can account for covariations within a dataset, it is difficult to interpret the logic behind clusters themselves. In such cases, feature projection helps to understand the underlying logic behind the formation of clusters.

Here we provide two examples of using feature projection to learn hidden patterns from a dataset. For brevity, we suffice to examples of autoencoder for two feature projection applications.

3.3.1. Example 1: Manual extraction of clusters

Here, we seek correlations between *input features* and *target metrics*. A scatter plot between solar gain and indoor air temperature will not provide any meaningful insights into the covariations (Fig. 11, left graph). This is further confirmed by their negligible coefficient of determination (R-squared) of 0.06.

The other option is to investigate parts of the data, rather than the whole dataset. We used autoencoder in section 3.2.1 to find groups of data that co-vary with indoor air temperature (Fig. 2, middle graph). One specific group of points labelled as 3 (Box #3), shows a strong covariation with solar gains (Table 2). We create another scatter plot between solar gain and indoor air temperature. This time however, instead of plotting the whole solar gain data against the whole indoor air temperature data, we only plot a small group of points, i.e. Box #3 (Fig. 11, right graph). The coefficient of determination of the new scatter is 0.29, which mean that during unoccupied hours, solar gains can explain ~ 30% of the variations of indoor air temperature.

As mentioned before, a building's response to indoor and outdoor variations is not constant. Therefore, using the entire dataset for seeking correlations could be uninformative or misleading. In this example, the operation of the HVAC system directly affects the indoor air temperature, and masks the correlation between solar gain and indoor air temperature as showed in Fig. 11.

3.3.2. Example 2: Understanding machine-learnt clusters

Next, we look at automated extraction of clusters by machine learning, and discuss the usefulness of principal components for understanding subsets of data. Here, we seek to explore the logic behind the formation of clusters, which will eventually help us decompose covariations between climatic/operational variables and indoor air temperature. As mentioned before, the conventional way of looking into covariations is by contrasting the whole dataset of target metrics against that of climatic or operational variables. For instance, the covariation between indoor air temperature and outdoor dry-bulb temperature would yield Fig. 12 (left graph), and rendering the clusters over the time-series plot of indoor air temperature would return Fig. 12 (right graph). Note that the R-squared value of this covariation is 0.38. Namely, the variations of outdoor dry-bulb temperature could explain that of indoor air temperature for 38% of the observations.

However, we argued that such assumptions could be an over-generalization of the building's performance. The R-squared value of 0.38 could be higher (or lower) at different periods of observation. Therefore, one should look into portions of the data, rather than the whole dataset. However, resorting to the cluster-labelled time-series plot of indoor air temperature (Fig. 12, right graph) would not provide much insight into the clusters and their characteristics, or how they correlate with indoor air temperature. Therefore, we contrast the indoor air temperature against the outdoor dry-bulb temperature using a scatter plot (Fig. 13). However, this time, we separately analyse the data from each cluster. The results show dissimilar performances for different clusters. For instance, Cluster 1 returns an R-squared value of 0.15, while Cluster 5 returns an R-squared value of 0.56. Both R-squared values greatly differ from that of the whole dataset.

We are interested to understand what specific characteristics of the subsets result in such difference. One of the great challenges of working with machine-learnt clusters is to understand the underlying logic behind their formation. Such information could potentially explain the difference between R-squared values in Fig. 13. To obtain an insight into the clusters, let us analyse how the clus-

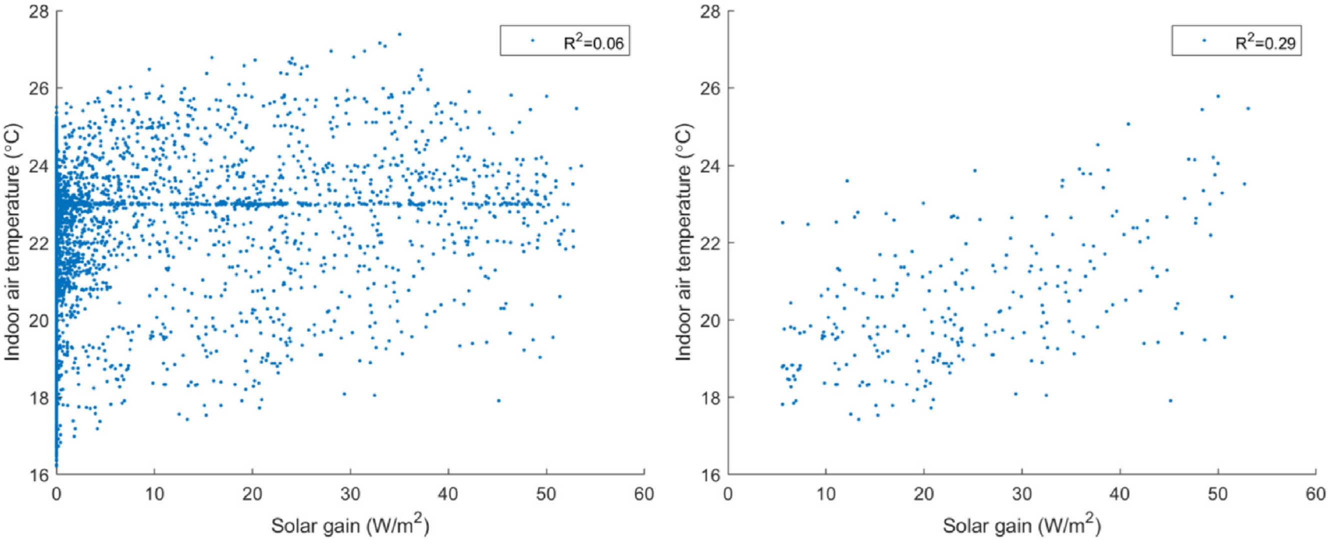

Fig. 11. Seeking correlations between solar gains and indoor air temperature. The whole dataset (left) and a portion of the dataset obtained from autoencoder preprocessing (right).

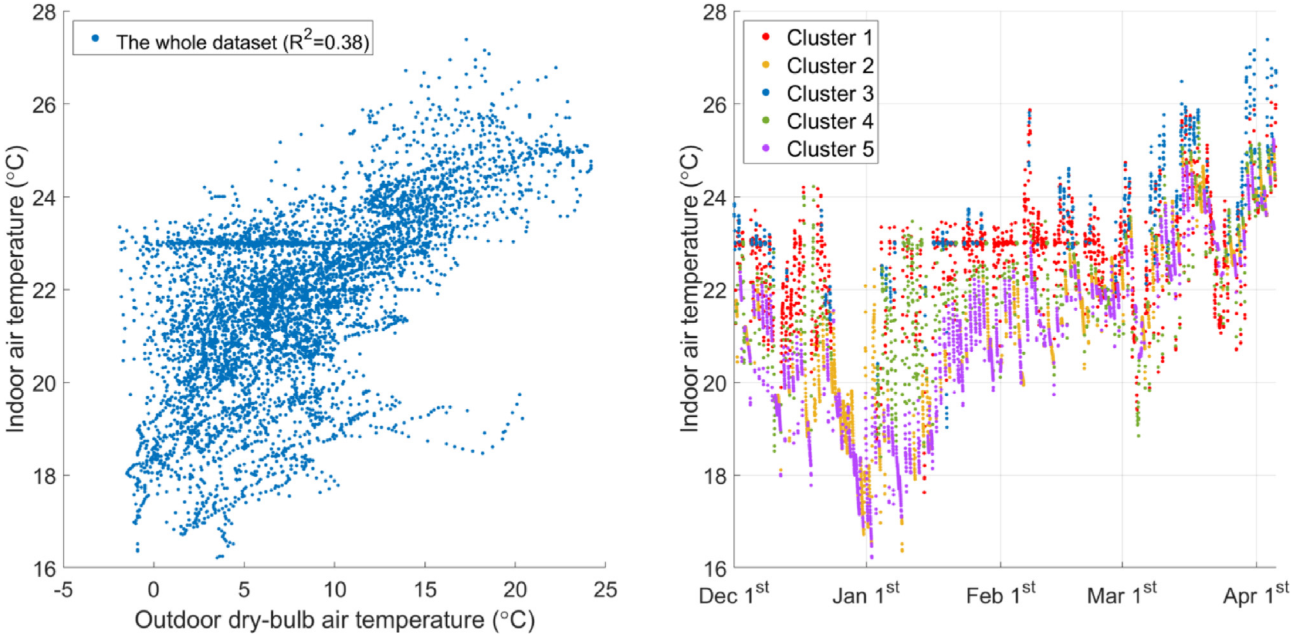

Fig. 12. The covariation between all observations of outdoor dry-bulb temperature and indoor air temperature (left), and rendering clusters over the indoor air temperature time-series plot (right).

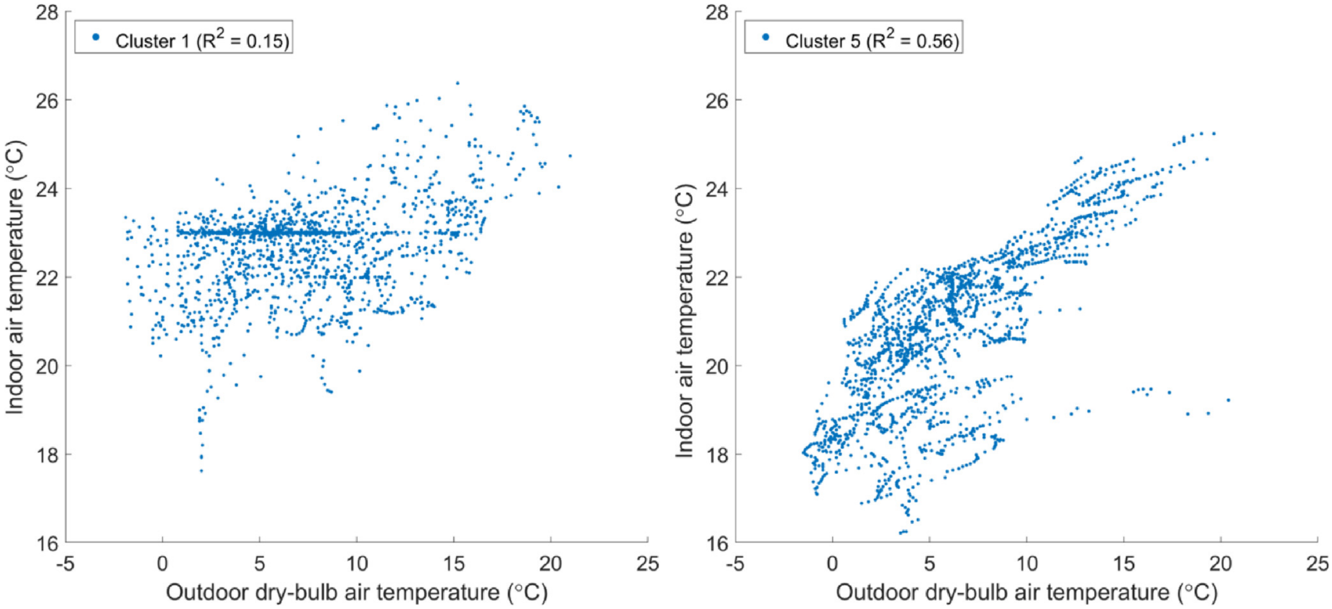

Fig. 13. Outdoor dry-bulb temperature versus indoor air temperature. A comparison between Cluster 1 (left) and Cluster 5 (right). For more details, see Fig. 10-middle graph.

ters are mapped on autoencoder's *2d hidden features* (see Fig. 10, middle graph). It is evident that clustering is mostly effective alongside the horizontal axis (Feature 1), rather than the vertical axis (Feature 2). Since autoencoder's Feature 1 is mainly correlated to operational variables (see Table 3), we can conclude that clusters differ due to operational patterns, rather than climatic patterns. Therefore, correlation between indoor air temperature and outdoor dry-bulb temperature is highly affected by the occupancy patterns. To better understand the effect of operation and climate on the indoor air temperature, we separately study each cluster (Fig. 14).

Cluster 5 shows a notable response to indoor air temperature. The cluster is highly dominated by climatic features, and shows no particular correlations with operational features (Fig. 14, right graph). This is evident from the scale of the horizontal axis, which spans from 0.975 to 1. Cluster 1 on the other hand, spreads alongside both operational and climatic axes, and displays a modest response to indoor air temperature (Fig. 14, left graph). By comparing Figs. 13 and 14, we observe that outdoor dry-bulb temperature can greatly affect indoor air temperature (R^2 = 0.56) during unoccupied hours. However, this effect can be heavily overshadowed by the presence of occupants (R^2 = 0.15). It is important to note that hidden features do not necessary show positive correlations with an input, and this phenomenon is often masked by the positive R-squared value. For instance in this study, Features 1 has a negative correlation with occupant density, which means that a

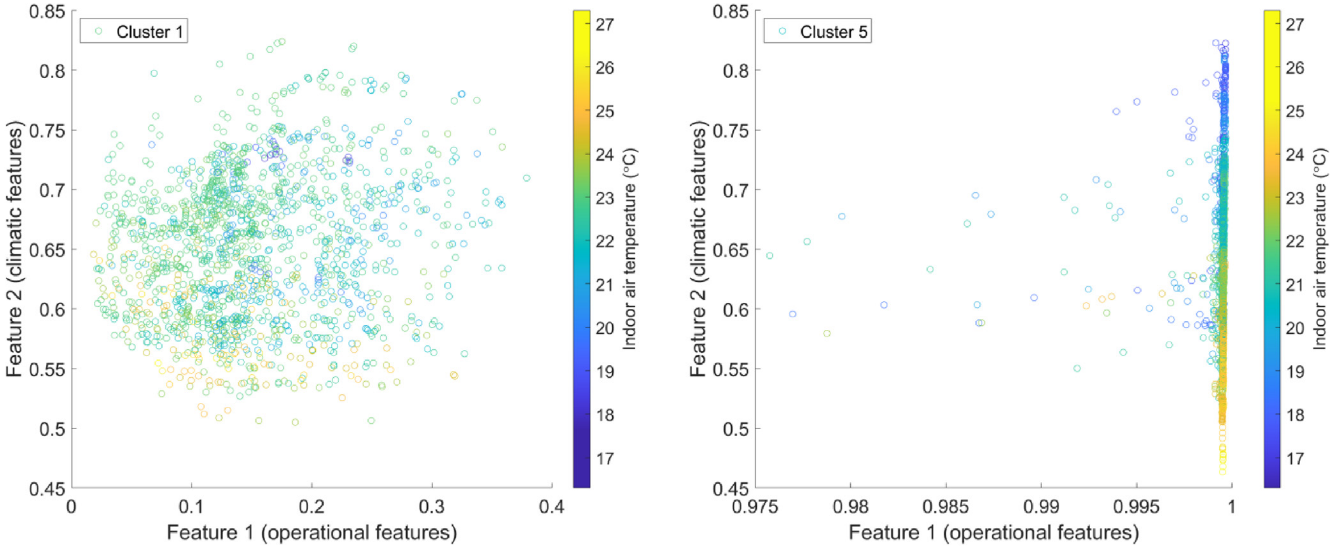

Fig. 14. Mapping indoor air temperature on autoencoder's 2d hidden features. A comparison between Cluster 1 (left) and Cluster 5 (right). For more details, see Fig. 4 (middle graph) and Fig. 10 (middle graph).

large value of Feature 1 corresponds to unoccupied periods. A similar pattern is observed for Feature 2 and climatic variables. Therefore, it is important to also check other metrics to ensure sufficient understanding of the covariations.

3.4. Issues with fine-tuning hyperparameters

Applying PCA on a dataset is relatively straight forward when there are no missing values and the number of observations

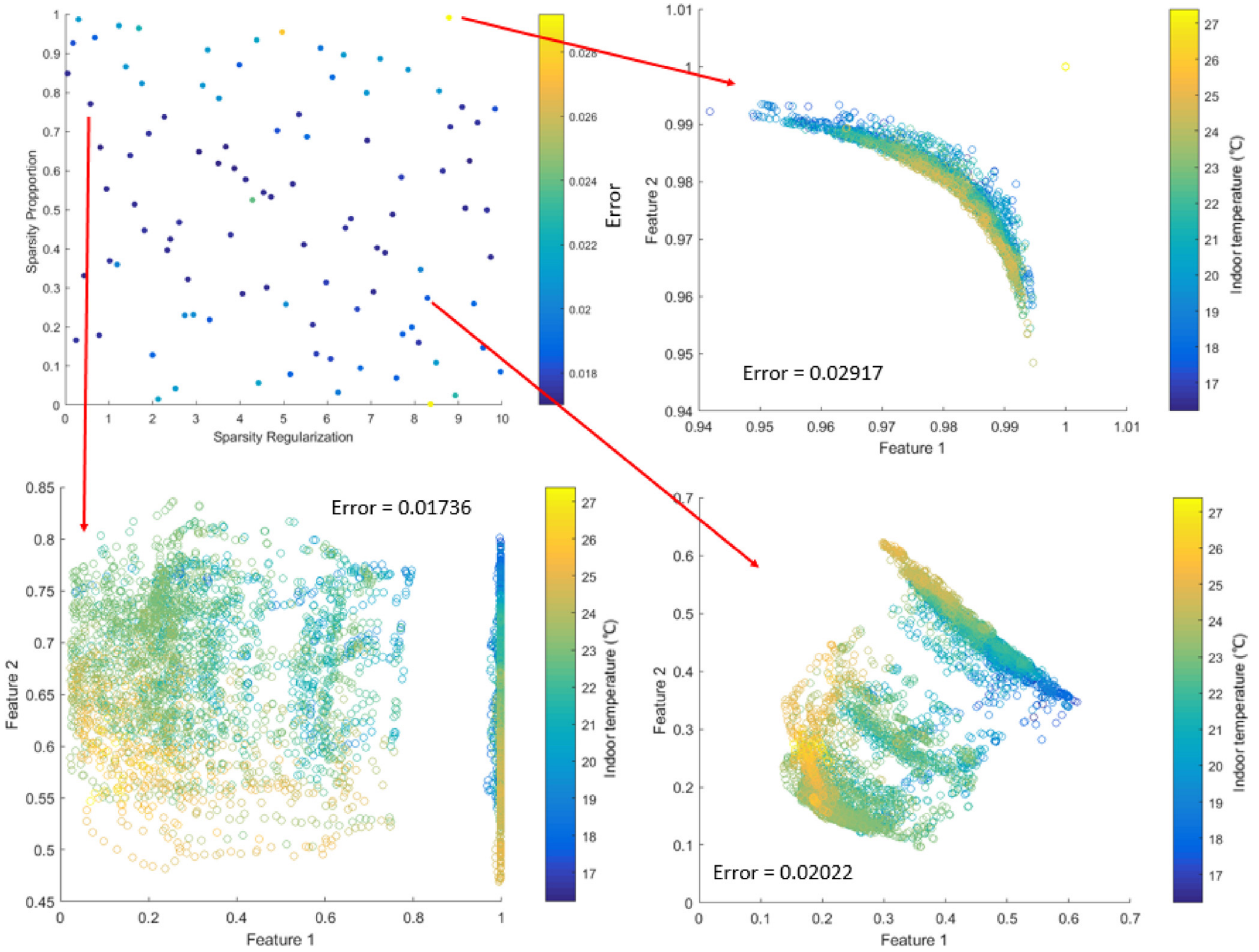

Fig. 15. Tuning autoencoder hyperparameters "Sparsity Proportion" and "Sparsity Regularization", and the effects on the reconstruction error (MSEsparse).

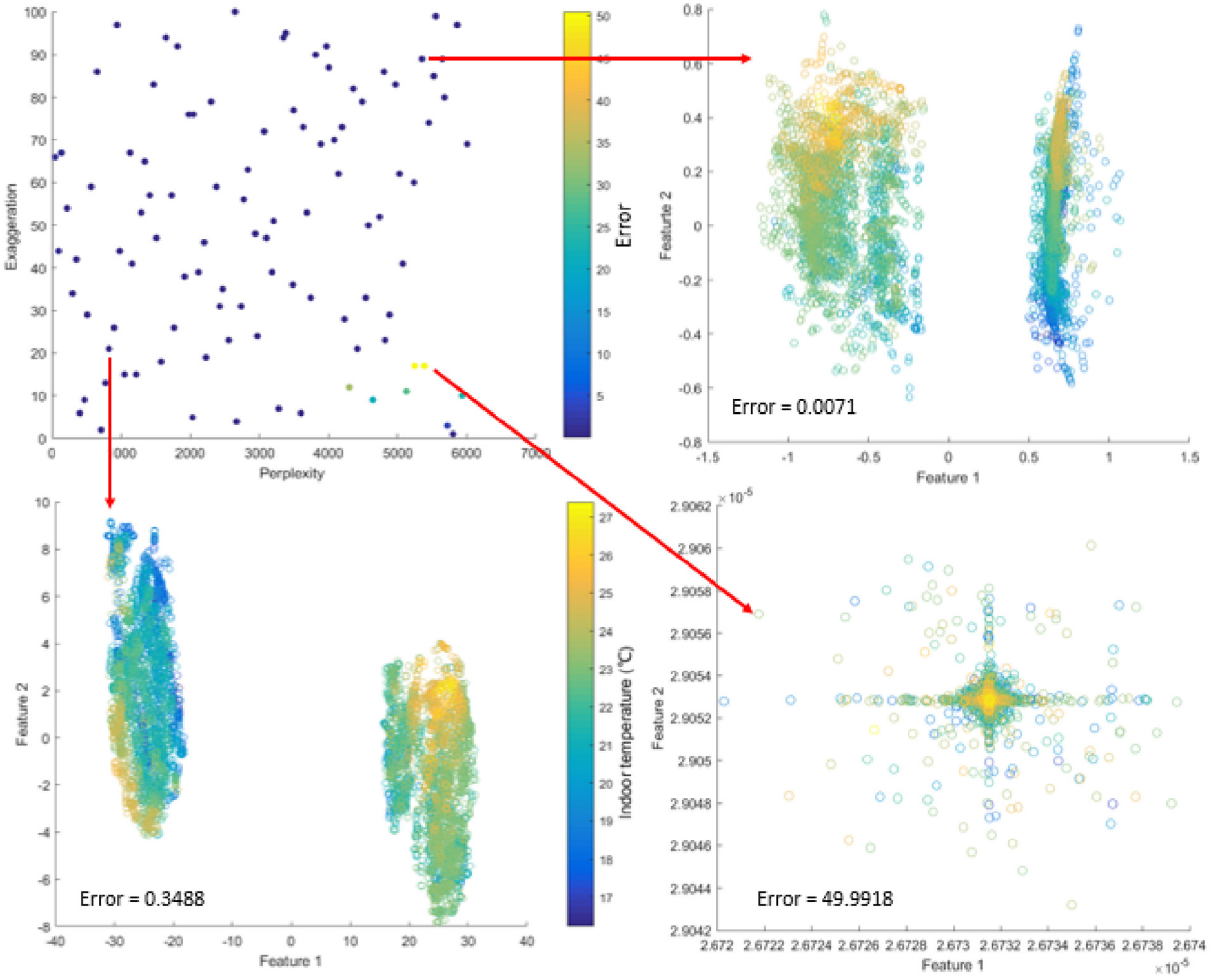

Fig. 16. Tuning t-SNE hyperparameters "Exaggeration" and "Perplexity", and the effects on the representation error.

exceeds the number of features. However, this is not particularly true concerning autoencoder or t-SNE. Both methods require careful fine-tuning to return optimal performance.

There are two parameters which strongly affect autoencoder's performance, i.e. "*sparsity proportion*" and "*sparsity regularization*". Sparsity proportion defines the fraction of observations to which a neuron fires, and therefore, encourages feature learning within specific parts of the dataset. Sparsity regularization penalizes weights that are close to zero or one, and consequently, prevents an autoencoder from overfitting into an identity matrix. Fig. 15 shows how autoencoder's reaction to a dataset can vary amidst changes to the hyperparameters. It is perceived that sparsity proportion values close to one or zero, are prone to worsen the autoencoder's performance.

Similarly, t-SNE's performance is strongly affected by two hyperparameters, i.e. exaggeration and perplexity. The suitability of t-SNE for representing well-separated clusters, has been previously established, yet, underlined that the hyperparameters play a significant role on its performance [43]. A grid search reveals that choosing low exaggeration and high perplexity values distort t-SNE's performance (Fig. 16). However, there is no specific rule of thumb for fine-tuning t-SNE's hyperparameters, therefore, it is advised to perform a systematic search to find a suitable set of values.

3.5. Summary

PCA, autoencoder and t-SNE are applied as linear and nonlinear dimensionality reduction algorithms to measurements collected from a teaching building. When only one feature is extracted from the dataset, all methods show a similar performance in finding highly relative parameters and amongst them autoencoder is found to be stronger in extracting features with regard to the R-squared value. When two features were combined together, equipment and solar gain are found highly related to the performance of the case-study building including indoor air temperature and heating energy consumption. In this case, heating consumption is found to be mostly sensitive to solar gain, but due to the strong influence of equipment gain any natural factor that significantly affects indoor air temperature is not identified. All methods show the feasibility of extracting information as a result of resorting to proper data compression and visualization techniques. In terms of t-SNE, instead of being a feature extracting tool, it may serve better for data visualization purposes when compared with PCA and autoencoder. Another drawback of t-SNE is that after training is complete, new sets of data cannot be encoded to the lower dimension, but rather, the training should be rerun from scratch with the new consolidated dataset. This may limit the applications of feature extraction with t-SNE.

The challenge of validating unsupervised learning methods, including dimensionality reduction is an ongoing research trend within the machine learning community [44]. In dimensionality reduction, there are no labels (ground truths) to assess the performance of a trained model. This does not mean that unsupervised models cannot overfit onto a dataset. Nevertheless, the notion of overfitting varies depending on the dimensionality reduction model. In some cases, it is possible to simply check the details of the trained model. For instance, in sparse autoencoders, overfitting means that the encoder has learned the identity matrix. In other cases, the unsupervised learning algorithm itself may have limitations. For instance, t-SNE does not learn an explicit function that maps input data to the latent feature space.

Dimensionality reduction becomes prone to overfitting when the size of the latent space (the target dimension) is large. Therefore, overfitting is not a great issue in this study, given that 1 or 2 latent features are extracted. However, we recommend assessing the vulnerability of dimensionality reduction methods in future studies.

4. Conclusion

Plot is regarded as the most commonly used method of data representation in different domains due to its comprehensibility and readability. However, in the field of building performance assessment, applications of plots were limited to simple representations without much pre or post-processing. In order to find more hidden information and limit information loss from a dataset, in this study, data representation was discussed by means of three dimensionality reduction algorithms (PCA, autoencoder and t-SNE). The dataset was recorded from December 2016 to April 2017 from a teaching building in Milan, Italy. Data was compressed into one or two features, so that patterns of correlation with heating energy consumption and indoor air temperature could be distinguished. When the dataset was reduced to a single dimension, all three methods displayed similar data extraction abilities, with t-SNE extracting more features relevant to indoor air temperature, and autoencoder mining more features relevant to energy consumption. However, when the dataset was compressed to two features, PCA and autoencoder surpassed t-SNE in distinguishing hidden patterns. In this case, lighting and equipment power, occupant density and to a lesser extent outdoor drybulb temperature were found to have the most powerful parameters influencing the indoor air temperature. Lastly, it is important to note that such level of understanding of the building's performance, including its reaction to the environment is obtained without the need for energy simulation and calibration, and solely by resorting to data mining.

Although information loss cannot be avoided at this stage, this approach gives sufficient legibility in the representation of data to decision makers. Obviously, the effect of data visualization by dimensionality reduction varies amongst different algorithms. In this paper, t-SNE showed a strong potential for visualizing two or three clusters, whilst, autoencoder displayed the best performance in distinguishing five subsets. As one of the aims of data visualization is to obtain hidden information, the distortion of data by t-SNE resulted in incomprehensible hidden patterns. Finally, it is important to note that solely resorting to the methods introduced in this study may not produce information that is adequately clear to comprehend, and therefore, further data mining after dimensionality reduction is inevitable.

Although this research successfully investigated the representation of building performance data through a case study, contrasted the performance of three dimensionality reduction algorithms, and provided hints on tuning the hyperparameters, limitations were inevitable as highlighted below.

- The data related to energy consumption analysis was mainly recorded from part of the heating period, which may not be adequate for a conclusive analysis. More data is recommended for obtaining better structures.
- The case-study of this study was a teaching building in Milan, Italy. Attempts with various building types such as office and residential buildings is highly recommended. In addition, the location of building could be taken into consideration as it heavily affects climatic features' structure.
- The correlation seeking was based on linear regression, which may result in information loss. Moreover, the chosen area of data gathering was selected randomly, and therefore the primary parameters found in PCA and autoencoder and t-SNE cannot match perfectly. Advanced algorithms such as semi-supervised neural networks are recommended for extracting more hidden information.

CRediT authorship contribution statement

Chunze Xiao: Data curation, Methodology, Software, Formal analysis, Visualization, Writing - original draft. **Fazel Khayatian:** Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing - review & editing, Supervision. **Giuliano Dall'O':** Resources, Conceptualization, Data curation.

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

We would like to thank Nicolò Di Blasi and Stefano Gironi from the University of Bocconi for sharing data as well as their expertise on the operation and energy management of the case study building.

References

[1] Iea.org., 2019. Buildings. [online] Available at: https://www.iea.org/topics/energyefficiency/buildings/.
[2] United Nations Environment Programme – Sustainable Buildings & Climate Initiative, 2009. Buildings and Climate Change: Summary for Decision Makers. p.9.
[3] ACEEE., 2018. Local Government Strategies for Achieving Energy Savings in Buildings. [online] Available at: https://aceee.org/local-policy/toolkit/savings-strategies-buildings.
[4] European Commission. Buildings - Energy Efficiency. [online] Available at: https://ec.europa.eu/energy/en/topics/energy-efficiency/buildings.
[5] UN Environment and International Energy Agency, 2017. Towards a zero-emission, efficient, and resilient buildings and construction sector. Global Status Report. p.14.
[6] WELL Building Standard v1, 2017. [ebook] New York: Delos Living LLC., p.107. Available at: https://www.wellcertified.com/en/content/well-building-standard.
[7] Urge-Vorsatz et al., Energy use in buildings in a long-term perspective, Current Opin. Environ. Sustain. 5 (2) (2013) 141–151.
[8] M. Pilgrim et al., Towards the efficient use of simulation in building performance analysis: a user survey, Build. Serv. Eng. Res. Technol. 24 (3) (2003) 149–162.
[9] Amasyali & El-Gohary, A review of data-driven building energy consumption prediction studies, Renew. Sustain. Energy Rev. 81 (P1) (2018) 1192–1205.
[10] Anders Haug, Frederik Zachariassen, Dennis van Liempd, The costs of poor data quality, J. Industr. Eng. Manage. 4 (2) (2011) 168–193.
[11] Redman, T., 2016. Bad Data Costs the U.S. $3 Trillion Per Year. [online] Harvard Business Review. Available at: https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year.
[12] P. De Wilde et al., Building simulation approaches for the training of automated data analysis tools in building energy management, Adv. Eng. Informat. 27 (4) (2013) 457–465.
[13] Jain et al., Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of

temporal and spatial monitoring granularity on performance accuracy, Appl. Energy 123 (C) (2014) 168–178.

[14] H.X. Zhao, F. Magoulès, Parallel Support Vector Machines Applied to the Prediction of Multiple Buildings Energy Consumption, J. Algorithms Computat. Technol. 4 (2) (2010) 231–249.

[15] Miller, Clayton & Schlueter, Arno, 2015. Forensically discovering simulation feedback knowledge from a campus energy information system. Doi: 10.13140/RG.2.1.2286.0964.

[16] O. Kimura et al., A prototype tool for automatically generating energy-saving advice based on smart meter data, Energy Efficien. 11 (5) (2018) 1247–1264.

[17] Thollander et al., International study on energy end-use data among industrial SMEs (small and medium-sized enterprises) and energy end-use efficiency improvement opportunities, J. Clean. Product. 104 (C) (2015) 282–296.

[18] Mena et al., A prediction model based on neural networks for the energy consumption of a bioclimatic building, Energy Build. 82 (2014) 142–155.

[19] Geyer, Singaravel, Component-based machine learning for performance prediction in building design, Appl. Energy 228 (2018) 1439–1453.

[20] T. Fleiter et al., The German energy audit program for firms—a cost-effective way to improve energy efficiency, Energy Efficiency 5 (4) (2012) 447–469.

[21] Yao, Steemers, A method of formulating energy load profile for domestic buildings in the UK, Energy Build. 37 (6) (2005) 663–671.

[22] D. Chakraborty, H. Elzarka, Advanced machine learning techniques for building performance simulation: a comparative analysis, J. Build. Perform. Simul. (2018) 1–15.

[23] Cecconi et al., Probabilistic behavioral modeling in building performance simulation: A Monte Carlo approach, Energy Buildings 148 (2017) 128–141.

[24] Y. Sun, L. Gu, C. Wu, G. Augenbroe, Exploring HVAC system sizing under uncertainty, Energy Buildings 81 (2014) 243–252.

[25] Gueta, Carmel, Quantifying the value of user-level data cleaning for big data: A case study using mammal distribution models, Ecol. Informat. 34 (C) (2016) 139–145.

[26] M. Pilgrim, D. Bouchlaghem, D. Loveday, M. Holmes, Towards best practice in building analysis data representation, Construct. Innovat. 2 (2) (2002) 117–130.

[27] A. Costa, M. Keane, J. Torrens, E. Corry, Building operation and energy performance: Monitoring, analysis and optimisation toolkit, Appl. Energy 101 (2013) 310–316.

[28] Capozzoli et al., Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings, Energy 157 (2018) 336–352.

[29] Ma et al., Building energy performance assessment using volatility change based symbolic transformation and hierarchical clustering, Energy Buildings 166 (2018) 284–295.

[30] C. Miller, Z. Nagy, A. Schlueter, Automated daily pattern filtering of measured building performance data, Automat. Construct. 49 (2015) 1–17.

[31] R. Edwards, J. New, L. Parker, Predicting future hourly residential electrical consumption: a machine learning case study, Energy Buildings 49 (2012) 591–603.

[32] Bellman, R.E., 1961. Adaptive control processes: a guide tour / by Richard Bellman., Princeton.

[33] Donoho, David, 2000. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. AMS Math Challenges Lecture. pp. 1-32.

[34] J.C. Lam, K.K. Wan, K. Cheung, L. Yang, Principal component analysis of electricity use in office buildings, Energy Buildings 40 (5) (2008) 828–836.

[35] Wei, W., Yan, H., Yizhou, W., Liang, W. Generalized Autoencoder: a neural network framework for dimensionality reduction. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2014), pp. 490-497.

[36] Goodfellow, Ian, Yoshua Bengio, Aaron Courville. 2016. "Deep learning. Book in preparation for MIT Press." URL¡ http://www. deeplearningbook. org 1.

[37] Hinton, Geoffrey E., Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science 313.5786 (2006): 504-507.

[38] Khayatian, F., 2018. Applications of Deep Machine Learning in Multi-Scale Building Energy Audit. Ph.D. Politecnico di Milano.

[39] L.J.P. van der Maaten and G.E. Hinton., 2008. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research, 9(Nov):2579-2605, 2008.

[40] OMD, Osservatorio Meteorologico di Milano Duomo (2017). http://www.meteoduomo.it/.

[41] Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics; 2007. p. 1027–1035.

[42] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.

[43] Wattenberg, et al., "How to Use t-SNE Effectively", Distill, 2016. Doi: 10.23915/distill.00002.

[44] Perry, Patrick O. "Cross-validation for unsupervised learning." arXiv preprint arXiv:0909.3052 (2009).