Applied Energy 211 (2018) 1123–1135

Contents lists available at ScienceDirect

Applied Energy

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

Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data

Cheng Fan[a], Fu Xiao[b], Yang Zhao[c,*], Jiayuan Wang[a]

[a] Department of Construction Management and Real Estate, Shenzhen University, Shenzhen, China
[b] Department of Building Services Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
[c] Institute of Refrigeration and Cryogenics, Zhejiang University, Hangzhou, China

HIGHLIGHTS

- An autoencoder-based ensemble method is developed for anomaly detection.
- Autoencoders can capture the intrinsic characteristics in building energy data.
- The performance of various autoencoder types and training schemes is compared.
- Methods are developed for performance evaluation without using anomaly labels.
- This study provides data-driven solutions to unsupervised anomaly detection.

ARTICLE INFO

Keywords:
Autoencoder
Unsupervised data analytics
Anomaly detection
Building operational performance
Building energy management

ABSTRACT

Practical building operations usually deviate from the designed building operational performance due to the wide existence of operating faults and improper control strategies. Great energy saving potential can be realized if inefficient or faulty operations are detected and amended in time. The vast amounts of building operational data collected by the Building Automation System have made it feasible to develop data-driven approaches to anomaly detection. Compared with supervised analytics, unsupervised anomaly detection is more practical in analyzing real-world building operational data, as anomaly labels are typically not available. Autoencoder is a very powerful method for the unsupervised learning of high-level data representations. Recent development in deep learning has endowed autoencoders with even greater capability in analyzing complex, high-dimensional and large-scale data. This study investigates the potential of autoencoders in detecting anomalies in building energy data. An autoencoder-based ensemble method is proposed while providing a comprehensive comparison on different autoencoder types and training schemes. Considering the unique learning mechanism of autoencoders, specific methods have been designed to evaluate the autoencoder performance. The research results can be used as foundation for building professionals to develop advanced tools for anomaly detection and performance benchmarking.

1. Introduction

Building operational performance has become one of the top concerns in achieving global sustainability. On the one hand, building operations are energy-intensive and contribute to approximately one-third of the world final energy consumption [1]. On the other hand, building operations have substantial energy saving potential considering the wide existence of equipment faults, energy-wasting occupant behaviors and improper control strategies. It is estimated that 16% of the energy consumed during building operations can be conserved through currently available energy management techniques [2]. One of the most promising solutions to tackling energy wastes during building operations is anomaly detection. Anomaly detection refers to the process of identifying rare observations in a data set [3]. It has been widely used in various industries, such as intrusion detection in network systems, fraud detection in financial transactions, and patient health monitoring in medical treatment [4]. In the building field, anomaly detection focuses on the detection and diagnostics of abnormal building operational patterns, which can be results of atypical operating behaviors, errors in sensing and transmission systems, equipment faults and energy-inefficient operating strategies [5]. It has been applied to three different levels, i.e., whole building level, subsystem level and

* Corresponding author.
E-mail address: youngzhao@zju.edu.cn (Y. Zhao).

https://doi.org/10.1016/j.apenergy.2017.12.005
Received 15 September 2017; Received in revised form 13 November 2017; Accepted 1 December 2017

component level [6]. The timely discovery of anomalies in building operations can be very helpful for building operation staff to understand building operating conditions, perform energy performance assessment and develop actionable measures for energy conservation.

Thanks to the development in information technologies, the real-time building operational performance can be well monitored and controlled through the building energy management systems (BEMS) or building automation systems (BAS). Massive amounts of building operational data are being collected and available for data analysis. It is therefore very promising to develop data-driven approaches to achieving reliable and robust anomaly detection. Based on the types of data analytics used, existing anomaly detection methods in the building field can be classified into two groups, i.e., supervised and unsupervised methods. The former adopts a model-based approach to anomaly detection. Supervised learning algorithms, such as support vector machines and multi-layer perceptions, are adopted for developing predictive models [7]. The model output is typically defined in two ways. The first is to use a variable to describe operating conditions as model output. In such a case, anomaly detection is performed based on the difference between predicted and actual values. Du et al. adopted neural networks to predict the supplied air temperature of the HVAC system given the supplied and returned chilled water temperature, the chilled water flow rates and etc. [8]. The difference between predicted and actual supplied air temperature was used for anomaly detection and various anomalies due to sensor biases, water valve stuck and controller faults were successfully discovered. Magoules et al. set the aggregated electricity consumption of building facilities, interior equipment, chillers, fans and pumps as the output for neural network models [9]. The method was successfully used to perform building-level anomaly detection. The other way of defining model output is to directly use a label stating whether an observation is an anomaly or not. In such a case, anomaly detection is transformed into a binary or multi-class classification problem. Zhao et al. developed a support vector machine-based method to detect anomalies in chiller operations [10]. A number of chiller operating parameters were used as model inputs and the model was trained using normal data alone. It was reported that anomalies at various severity levels could be successfully identified. Guo et al. applied a back-propagation neural network to identify anomalies in a variable refrigerant flow air conditioning system [11]. The model output was set as labels stating whether observations were normal or faulty. Despite of the effectiveness of model-based methods, their practical values are usually limited. The reasons are: (1) Obtaining high-quality training data can be time-consuming; (2) Obtaining the label (or ground truth) on whether an observation is abnormal or not can be costly and sometimes infeasible.

By contrast, unsupervised anomaly detection methods are more promising for practical applications, as they do not require anomaly labels. Existing methods can be further classified into two types, i.e., statistical methods and unsupervised data mining methods. Statistical methods make statistical assumptions on the underlying data distribution (e.g., Gaussian normal distribution), based on which scores are calculated for anomaly detection. One prominent example is the generalized extreme studentized deviate (GESD)-based method. Previous studies have demonstrated the usefulness of GESD-based methods in identifying anomalies in building energy consumption profiles [12,13]. The method typically involves a feature extraction step to represent high-dimensional data as low-dimensional features. The GESD algorithm is then applied on features for anomaly ranking [14]. Other statistical methods include the nearest neighbors and principal component analysis-based methods [15,16]. Recently, unsupervised data mining techniques have gained increasing interests in anomaly detection due to their excellence in handling massive and complicated data sets [17,18]. The most widely used unsupervised data mining techniques in the building field include clustering analysis and association rule mining (ARM) [18]. The original intention of clustering analysis is to group similar observations into one cluster. Advanced clustering algorithms have been designed for anomaly identification. As illustrated in [19], the density-based spatial clustering of applications with noise (DBSCAN) algorithm could effectively identify anomalies in building energy consumption data. Association rule mining (ARM) aims to extract significant associations among data variables. The knowledge discovered can therefore be applied for anomaly detection, e.g., directly identifying rules on energy waste patterns or detecting observations not fulfilling normal associations [20,21]. Cabrera and Zareipour used ARM to identify abnormal energy waste patterns in lighting systems of educational buildings [22]. Fan et al. developed an anomaly detection engine based on normal associations discovered from massive operational data [23].

To summarize, the requirement of labeled high-quality training data has imposed great constraints on the applicability of supervised anomaly detection methods. Unsupervised anomaly detection is more flexible for practical applications. The main limitations of existing unsupervised anomaly detection methods are: (1) The anomaly detection performance and computational efficiency can be degraded dramatically when applying to big data. For instance, statistical methods are not scalable to large-scale data and they are subject to stringent mathematic assumptions, which may not be fulfilled by real-world high-dimensional data. Some unsupervised data mining techniques have been used to enhance the effectiveness and efficiency in analyzing big data. Nevertheless, the associated post-mining workload can be overwhelming, e.g., selecting useful and non-redundant association rules describing normal or faulty working conditions can be very time-consuming [24,25]. (2) The performance of existing unsupervised methods relies heavily on features used. Currently, features for anomaly detection are selected or constructed based on domain expertise or simple statistics (e.g., the mean and standard deviation of a numeric variable). There is a lack of data-driven methods to automate the feature generation process for generalization purposes. More advanced methods are desired to enhance the performance and applicability of unsupervised anomaly detection in the building field.

One promising solution to these limitations is the autoencoder. An autoencoder adopts the neural network architecture to perform unsupervised learning, where the model input and output are set identical. The rapid development in the deep learning community has provided various techniques for analyzing different types of data (e.g., cross-sectional or temporal data) and training models with advanced architectures (e.g., deep convolutional autoencoders) [26]. More importantly, autoencoders enable a data-driven approach to high-level feature extraction, which can be used to tackle the most challenging task in unsupervised anomaly detection, i.e., feature engineering [27].

To the best of the authors' knowledge, there is a lack of studies to systematically examine the potential of different types of autoencoders in the unsupervised anomaly detection of building operational data. This study is performed to fill this knowledge gap. More specifically, an autoencoder-based ensemble method is proposed for detecting anomalies in building energy data. The autoencoder ensemble is developed considering different autoencoder architectures and training schemes. In addition, novel methods are developed to evaluate the autoencoder performance without the use of anomaly labels. The paper is organized as follows: Section 2 introduces the basics on autoencoders. Section 3 describes the research methodology, including the research outline, the development of autoencoder-based ensembles for anomaly detection and the proposed performance evaluation methods. Section 4 presents a case study. The research results are shown and discussed in Section 5. Conclusions are drawn in Section 6.

2. Basics on autoencoders

2.1. The general autoencoder architectures

Autoencoders can be regarded as a special form of neural network designed for unsupervised learning [28]. The learning process is

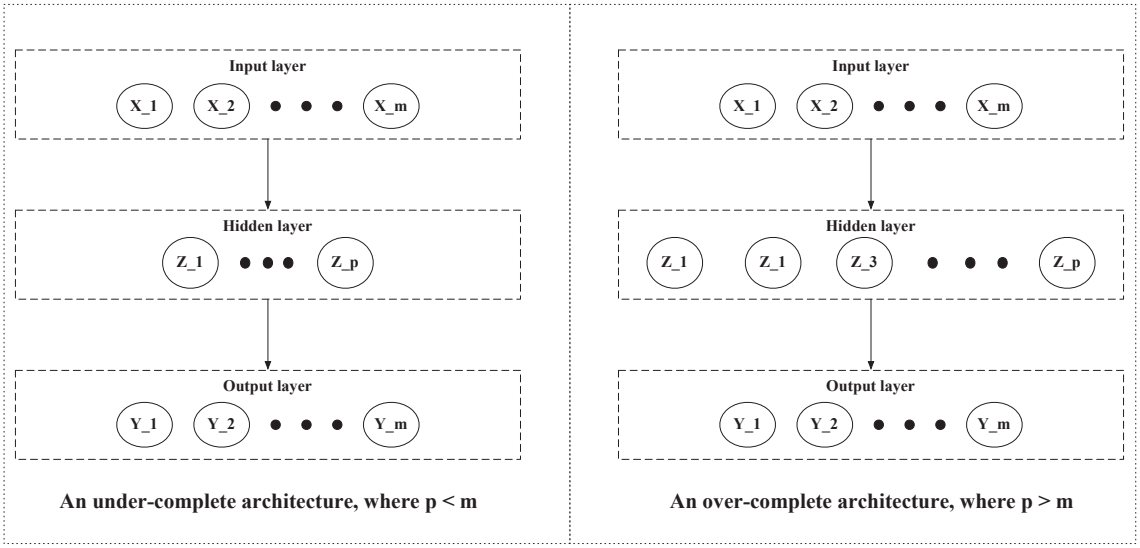

Fig. 1. The general autoencoder architectures.

unsupervised since there is no label variable. The input (i.e., denoted as X) and output (i.e., denoted as Y) are set identical with a dimension of n and m, where n is the number of observations and m is the variable number. An autoencoder consists of an encoder and a decoder. The encoder transforms the input data into high-level features (i.e., denoted as Z), while the decoder tries to reconstruct the input data using high-level features. Mathematically, an encoder learns a mapping of $Z = f_\theta(X) = S(WX + b)$, where W and b are matrices of weights and biases respectively, and S represents activation functions (e.g., sigmoid or hyperbolic tangent). Similarly, a decoder learns a mapping of $Y' = f_{\theta'}(Z) = S(W'Z + b')$. An autoencoder is trained to minimize the reconstruction residuals between Y and Y', which is typically evaluated using mean squared errors or cross-entropy losses [29]. It is noted that by setting p equal to or larger than m, a trivial identity mapping can be used to achieve perfect reconstruction [29]. Therefore, training constraints are usually specified in the autoencoder architecture to learn meaningful high-level features. As shown in Fig. 1, two autoencoder layouts are commonly used: (1) an under-complete or a bottleneck layout where p is smaller than m; (2) an over-complete layout where p is larger than m. The under-complete layout learns a compressed representation of X while the over-complete layout learns a sparse representation of X. The sparse over-complete representation can be regarded as a special case of under-complete representation, as the majority of hidden neurons are forced to be zeros. Some studies have reported better feature learning capability using the over-complete layout [29]. However, the number of model parameters may grow significantly and require more data to ensure the reliable and robust training.

2.2. Types of autoencoders

A variety of autoencoder algorithms have been developed to handle different types of problems. The basic version of an autoencoder consists of three fully connected layers, i.e., one input layer, one hidden layer and one output layer. The reconstruction ability can be improved by introducing more hidden layers and hidden neurons. However, the feed-forward fully connected autoencoders cannot easily capture the structural dependency, such as temporal dependency in 1-dimensional (i.e., 1D) time series and spatial dependency in 2-dimensional (i.e., 2D) image data [30]. Convolutional neural networks are hierarchical models which adopt convolution and pooling operations to capture these structural dependencies. It has been utilized as the primary tool for 2D image classification and 1D signal processing [31]. Convolutional neural networks can effectively reduce the number of model parameters by limiting the neuron connections with input data. Similar configurations can be applied for developing convolutional autoencoders (CAE). Considering that building operational data are in essence time series data, CAE may provide more reliable performance than fully connected AEs. Another popular type of neural networks, i.e., the recurrent neural network, is specially designed to analyze sequential data. Cycles are designed between nodes in recurrent neural networks, based on which dynamics in sequential data are captured. Recurrent autoencoders can achieve high-quality reconstruction of time series data. We refer readers to [32,33] for detailed descriptions on convolutional neural networks and recurrent neural networks.

2.3. Denoising training schemes

Adding noises to input data X is a special training scheme to avoid the learning of trivial identity mapping. The corrupted input data is denoted as X'. In such a case, the autoencoder is trained to extract useful features for denoising, i.e., reconstruct X based on corrupted input X'. This training scheme has proved to be very useful in learning reliable and robust high-level features. There are generally three types of ways to add noises [29]. The first considers additive isotropic Gaussian noises. It is suitable for real-valued inputs and noises are added to each input based on a Gaussian normal distribution. The second is called salt-and-pepper noise, where a fraction of input values are randomly selected and set as either zeros or ones. It is a natural choice for binary or near binary input. The third is called masking noise. More specifically, a fraction of input values is randomly selected and their true values are masked using zeros.

Considering that building operational data consists of a considerable amount of missing values and unreliable measurements, this study specifically investigates the performance of denoising autoencoders under different levels of masking noises. To illustrate, Fig. 2 presents an example of denoising autoencoder training using masking noises. The input data X is a matrix with two rows and five columns, which means there are two observations each with five variables. The masking noise parameter v is set to be 0.2, indicating that one variable (i.e., $0.2 \times 5 = 1$) is randomly selected and set as zero for each observation. As an example, the second and fourth variables are set as zeros in the first and second observations in X respectively. A denoising autoencoder is then developed using the corrupted input X' as input and the original input X as output.

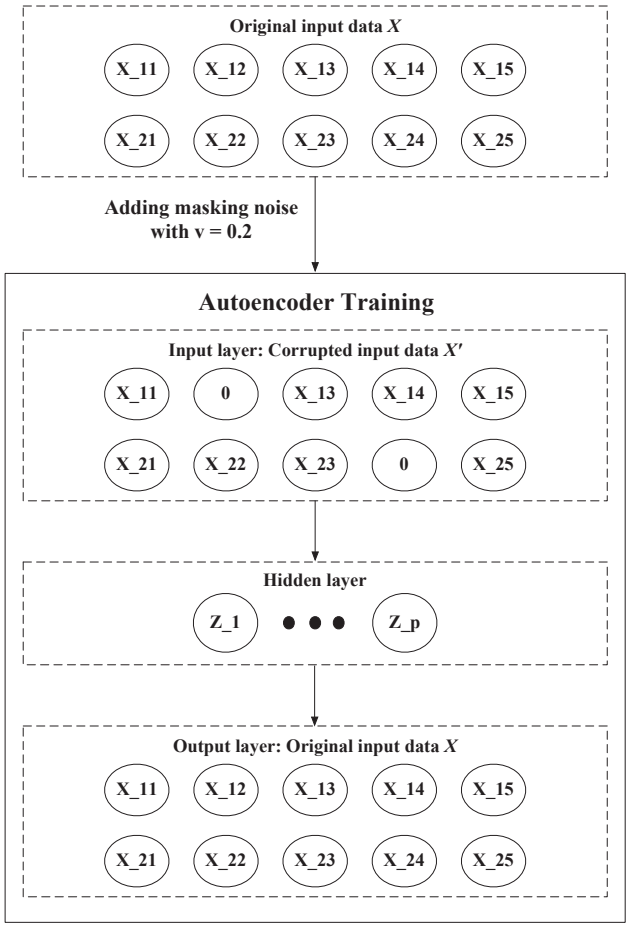

Fig. 2. An example of denoising autoencoder using masking noises.

3. Research methodology

An autoencoder-based ensemble method is proposed to detect anomalies in building energy consumption data. As shown in Fig. 3, the method consists of four steps to achieve unsupervised anomaly detection in building energy data. Two evaluation methods, which do not require the availability of anomaly labels, are developed to compare the performance of different autoencoders.

3.1. Data exploration

The first step is data exploration, which aims to capture the intrinsic characteristics in building energy consumption data. Two tasks are involved, i.e., identification of dominant periods and influential exogenous variables. Building energy consumption data are essentially time series data with unknown periodicities. To ensure the efficiency and effectiveness in anomaly detection, it is conventional to transform the time series into a matrix of subsequences. Meaningful anomalies can then be detected by finding atypical subsequences. Previous studies mainly adopted domain expertise to determine the subsequence length [34]. To minimize the dependency on domain expertise and to maximize the possibility of discovering previously unknown knowledge, this study adopts a data-driven method, i.e., the spectral density estimation method, for period identification. Both parametric and non-parametric methods are available for spectral density estimation. Parametric methods rely on the parameters of time series models (e.g., autoregressive and moving average models) to estimate spectral density. Non-parametric methods calculate the spectral density based on autocorrelation function and the Fourier transformation. It is reported that non-parametric methods might suffer from poor frequency resolution and spectral leakage problem, making it inconsistent to identify the true periodic components in time series [35]. Therefore, the parametric method based on autoregressive models is adopted in this study.

The other important task in data exploration is to identify the most influential exogenous variables to building energy consumption. Some exogenous variables, such as time variables (*Month*, *Day Type*, *Hour*), may greatly change the behavior of building operations. Considering that an observation may be identified as an anomaly in one context but not in another, it is helpful to integrate the information of influential exogenous variables into the anomaly detection process. Such type of information is used as conditional information for autoencoder development. In the data exploration step, the decision tree method is used to identify the most influential exogenous variables to building energy consumption considering its convenience in model visualization and knowledge interpretation.

3.2. Generation of building energy consumption subsequences

Based on the results of period identification, the long time series of building energy consumption are segmented into subsequence matrix. Subsequences are created using a sliding window with a size equal to the most dominant period identified. In addition, different masking noise levels are used to generate corrupted subsequences. For instance, a masking noise level of 5% means that for each row in the subsequence matrix, 5% of the values are randomly selected and set as zeros.

3.3. Autoencoder-based ensemble for anomaly detection

Individual models or algorithms usually have limitations and the resulting performance may vary greatly in different scenarios. Ensemble learning is an effective approach to enhancing the result reliability and robustness. It is originally used in predictive modeling, where the prediction performance can be improved significantly by combining results of a number of models [36]. The recent research on ensemble learning has proved its usefulness in unsupervised anomaly detection [37].

In this study, a number of autoencoders are developed using different architectures and training schemes. In terms of autoencoder architectures, the basic fully connected feed-forward architecture and 1D convolutional architecture are used. Training schemes differ in two aspects, i.e., whether denoising training and conditional information is used or not. An ensemble is developed based on these individual autoencoders.

Two tasks are involved in constructing ensembles for anomaly detection. The first is to calculate an anomaly score for each observation using a base autoencoder. The anomaly score for each subsequence is derived from the corresponding sequence of reconstruction residuals. The root mean squared error is adopted for reconstruction loss calculation, i.e., $\sqrt{\frac{\sum_{i=1}^{n}(x_i - x_i')^2}{s}}$, where s is the subsequence length. Considering that the anomaly scores generated by different autoencoders may have different scales, the max-min normalization $\left(\text{i.e. , } Score_i' = \frac{Score_i - Score_{min}}{Score_{max} - Score_{min}}\right)$ is applied to normalize the anomaly scores generated by each autoencoder before score aggregation. The resulting scores range from zero to one and can be regarded as possibilities of being an anomaly. The mean-based aggregation method is then applied to calculate the final anomaly score for each subsequence. Top candidates are detected by sorting the final anomaly scores in a descending order. Due to the lack of anomaly labels, it can be very difficult to define the anomaly detection threshold. Two general approaches are available to use: (1) specify the cut-off values based on domain expertise or anomaly score visualization; (2) specify the threshold based on common assumptions of anomalies, e.g., 5–10% of the data are anomaly candidates [3,38].

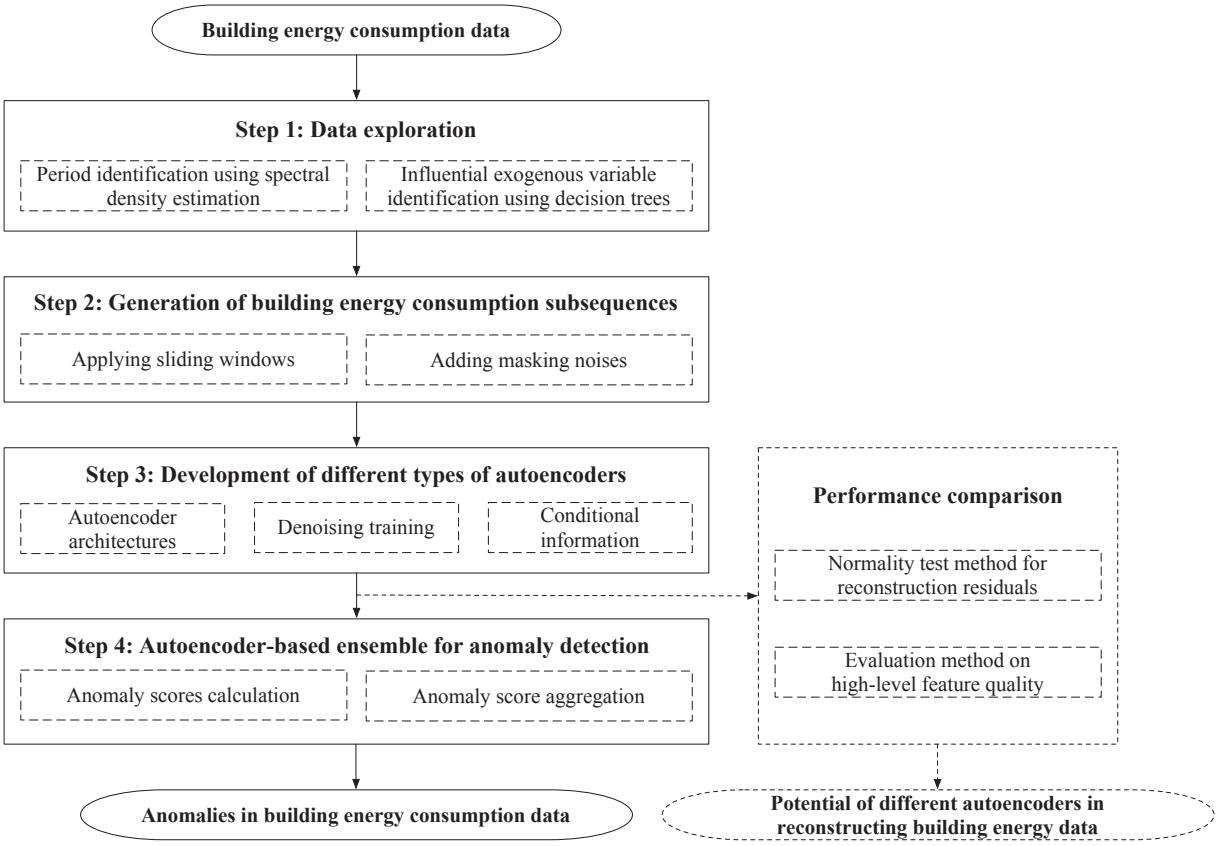

Fig. 3. Research outline.

3.4. Performance evaluation

One of the most critical challenges in unsupervised anomaly detection is that one cannot easily evaluate the performance due to the lack of anomaly labels. As a solution, this study proposes two methods to indirectly evaluate the autoencoder performance, i.e., (1) performing normality test to check the reconstruction residual distribution; (2) performing a supplementary classification task to evaluate the quality of high-level features extracted by autoencoders.

Firstly, assuming that the autoencoders are well developed, the residuals between reconstructed and actual values in normal data can be regarded as random errors. Previous studies have shown that the distribution of random errors in most real-world processes can be modeled using a normal or near-normal distribution with a mean of zero [39]. Therefore, the validity of autoencoders can be partially assessed by checking whether the reconstruction residual distribution is normal or not. Among many test methods, the Shapiro-Wilk test, which is based on the correlation between the data and the corresponding normal scores, is reported to be the most reliable one [40]. It is therefore selected in this study. Considering that the recommended sample size for normality test is less than 50 [41], the normality test is carried out separately for each reconstruction residual sequence. Given a significant level α, the percentage of residual sequences passing the normality test is recorded for performance comparison. The percentage is expected to be higher for autoencoders with better reconstruction ability.

Secondly, the autoencoder performance can be evaluated based on the quality of high-level features extracted [29]. To investigate the quality of high-level features, a supplementary classification task is designed. Even though the ground truth of whether an observation is an anomaly or not is unknown, other available information (e.g., *Hour* and *Day Type*) can be used as labels for classification. In such a case, a predictive model is developed using the high-level features as inputs. It is expected that better prediction performance can be obtained using features with better quality.

4. Case study

4.1. Description of building operational data

The methodology is applied to analyze the building operational data retrieved from an educational building in Hong Kong. The building mainly includes offices for university staff, classrooms for students, and a data center for computing devices. It has an approximate gross floor area of 11,000 m^2 while 8500 m^2 are air-conditioned. The building operational data were recorded using a collection interval of 30-min. The data recorded in 2015 are used for analysis. In total, 113 variables are recorded and can be generally classified into four types: (1) time variables (i.e., *Month*, *Day*, *Hour*, *Minute* and *Day type*); (2) outdoor variables (e.g., outdoor dry-bulb temperature and relative humidity); (3) operating parameters of the chiller plant (e.g., the temperatures and flow-rates of chilled water and condenser water); (4) energy variables (e.g., the total building cooling load and electricity consumption of the chiller plant).

4.2. Identification of intrinsic characteristics in time series data

This study focuses on detecting anomalies in building energy data and the total building cooling load is selected as the target. As described in Section 3.1, the first task in data exploration is to identify intrinsic periodicities in time series, based on which the length of subsequences is determined. Fig. 4 presents the spectral density estimated using the autoregressive model-based parametric method. The red[1] cross

[1] For interpretation of color in 'Figs. 4, 8, and 12', the reader is referred to the web version of this article.

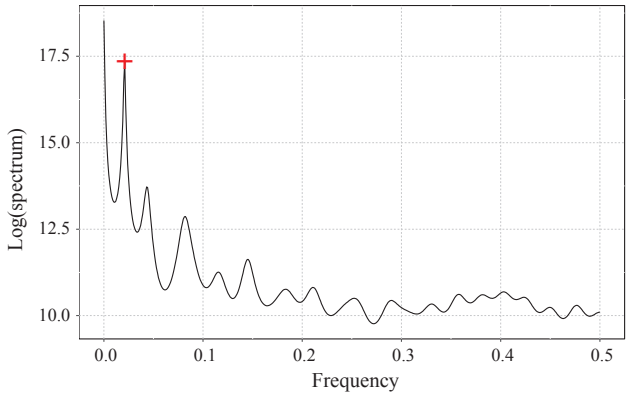

Fig. 4. Spectral density of the total building cooling load.

represents the most significant frequency (i.e., 0.021) in the time series of building cooling load. It corresponds to a period of 47.6 $\left(\text{i.e. , } \frac{1}{0.021}\right)$. Since the data are recorded at a 30-min time interval, the period actually indicates a daily seasonality, i.e., $\frac{47.6}{2} = 23.8$-h. Therefore, the subsequence length is defined as 48 (i.e., 24-h). Overlapping subsequences are generated considering an incremental time step of 2 (i.e., 1-h). In total, 8497 subsequences are generated, resulting in a subsequence matrix with 8497 rows and 48 columns.

The second task in data exploration is to discover influential variables to building cooling load. The main aim is to find suitable variables as conditional information for the reliable and robust anomaly detection. The decision tree method is adopted to visualize the relationships between the aggregated daily building cooling load and time variables. As shown in Fig. 5, two time variables, i.e., *Month* and *Day Type*, are identified as influential to daily building cooling load. Node 1 selects *Month* as the splitting variable with two groups of values, i.e., {*1*, *2*, *3*, *12*} and the others. The former corresponds to the cooler and less humid seasons in Hong Kong while the latter are hotter and more humid seasons. Node 3 indicates that building cooling load is also affected by *Day Type*. The results are in accordance with domain expertise, as building cooling load is expected to be small on Sundays due to the lack of teaching and working activities. From the perspective of anomaly detection, integrating the information brought by these two exogenous variables can be helpful to enhance the anomaly detection performance. For instance, small building cooling loads are normal on Sundays during cold seasons, but are more likely to be anomalies when observed on weekdays during hot seasons. Considering that the aim of this study is not only focused on autoencoder development but also performance evaluation, only *Month* is used as conditional information for autoencoder development and *Day Type* is used as the output for the supplementary classification task designed for performance evaluation. The consideration behind is to ensure the credibility of classification results by not introducing the *Day Type* information into the high-level features.

4.3. Development of autoencoder-based ensembles

As summarized in Table 1, 20 individual autoencoders are developed considering different architectures and training schemes. Fully connected feed-forward and 1D convolutional architectures are considered and the resulting autoencoders are denoted with '*Basic*' and '*Conv*' respectively. The *Month* is integrated into the autoencoder development process as conditional information. If conditional information is not used, the autoencoder is denoted with '*-A*' and otherwise '*-B*'. One-hot encoding is used to transform each value in *Month* into a vector with 12 values, i.e., all values are set zeros except one corresponding to a certain month. Five masking noise levels are considered, i.e., 0%, 5%, 10%, 15% and 20%. The resulting autoencoders are denoted with '*-1*', '*-2*', '*-3*', '*-4*' and '*-5*' respectively. Considering a round operation for removing decimals, masking noise levels of 0%, 5%, 10%, 15% and 20% indicate that 0, 2, 5, 7 and 10 out of 48 values are randomly selected and set as zeros for each subsequence.

The data set contains 355 complete days and therefore, 355 daily subsequences are used for testing and performance evaluation. The rest of the subsequences are used for autoencoder training. The building cooling load data are transformed using max-min normalization $\left(\text{i.e. , } Y_i' = \frac{Y_i - Y_{min}}{Y_{max} - Y_{min}}\right)$ and therefore, the resulting building cooling loads range between 0 and 1. Such normalization enables faster training, reduces the chances of getting stuck in local optima, and can preserve exactly all relationships in the original data [42]. The activation function used at the output layer is the sigmoid function, as it matches the range of normalized data. To ensure the reliability in parameter tuning, the sizes of autoencoders are limited. Each autoencoder is trained using Adam optimizer with a learning rate of 0.001, 50 epochs and a mini-batch size of 10. It is worth mentioning that better reconstruction performance can be achieved through parameter tuning. The parameters are set identical in this study for the purpose of performance comparison. The detailed configurations of autoencoders are shown in Figs. 6 and 7. The same configuration is used for each type of architecture and the only difference lies in the input data, i.e., whether conditional information and corrupted inputs are used or not.

As described in Section 3.3, reconstruction losses are calculated through root mean squared errors, i.e., $\sqrt{\frac{\sum_{i=1}^{48}(x_i - x_i')^2}{48}}$, where X_i is the normalized building cooling load at time step i and X_i' is the reconstructed value at time step i. The reconstruction losses generated by each autoencoder are normalized through max-min normalization. The final ensemble score for each subsequence is aggregated based on the mean operation.

5. Results and discussions

This section presents the research results from two perspectives. Firstly, the anomaly detection results are illustrated. Secondly, the performance of different types of autoencoders is carefully examined, providing insights into the applicability of different autoencoders in practice.

5.1. Anomaly detection results

As described in Section 4.3, top anomalies can be identified by sorting the ensemble anomaly scores in a descending order. The threshold is determined based on the common assumption that 5% of the data are anomaly candidates [38]. It is found that the anomalies identified generally fall into three categories: (1) anomalies due to atypical or rare operations; (2) anomalies due to transient operations; (3) anomalies due to improper control strategies. Figs. 8–10 present examples from these three categories. In general, it is observed that the use of conditional information *Month* helps to smooth the reconstructed profiles. The reconstructed profiles can be used as better benchmarks considering the influence of seasonality. In addition, the increase in masking noise levels does not necessarily lead to higher reconstruction losses, and all the reconstructed profiles can capture the actual trend. It indicates that autoencoders can be used to reconstruct building operational data with missing values or transient measurements.

More specifically, Fig. 8 presents an example which corresponds to the daily operation on July 2nd, 2015. The actual normalized building cooling load is shown in red solid line, while the reconstructed profiles generated by different autoencoders are shown in other formats. The main mismatches between actual and reconstructed building cooling loads take place between 0 a.m. and 8 a.m. During this period, two unusual spikes in building cooling load are observed. Such type of operations can be related to the pre-cooling control logic or a sudden increase at the demand side. It should be noted that such anomalies do not necessary represent faulty operations. Rather, they should be

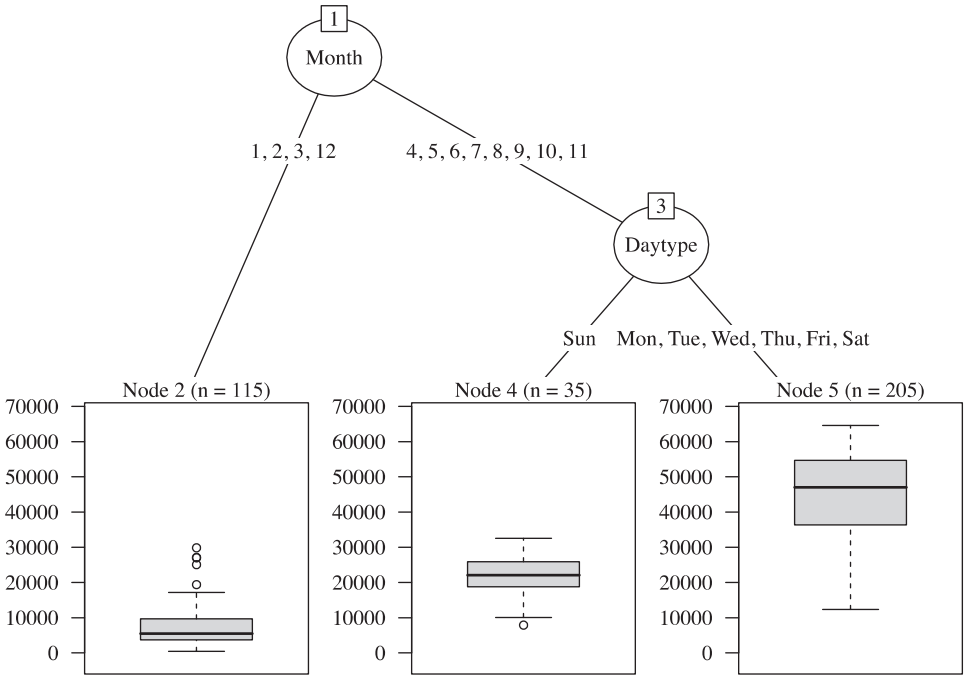

Fig. 5. Influential variables identified to building cooling load.

Table 1
Summary of autoencoder development.

Autoencoders	Architectures	Conditional info.	Masking noises
Basic-A-1	Fully connected feed-forward	Without conditional info.	0%
Basic-A-2			5%
Basic-A-3			10%
Basic-A-4			15%
Basic-A-5			20%
Basic-B-1		With conditional info.	0%
Basic-B-2			5%
Basic-B-3			10%
Basic-B-4			15%
Basic-B-5			20%
Conv-A-1	Convolutional	Without conditional info.	0%
Conv-A-2			5%
Conv-A-3			10%
Conv-A-4			15%
Conv-A-5			20%
Conv-B-1		With conditional info.	0%
Conv-B-2			5%
Conv-B-3			10%
Conv-B-4			15%
Conv-B-5			20%

treated as atypical operations or rare events. The discovery of these anomalies can help building professionals to better understand the actual characteristics in building operations.

Fig. 9 presents an example taking place on July 14th, 2017. The abnormality mainly comes from the huge spike at 6 p.m. In such a case, these spikes represent transient operations and are usually observed when multiple equipment in chiller plants are switched on at the same time. It is observed that the reconstructed profiles have smoothing effects and can be used as benchmarks to describe non-transient operations. Again, it should be noted that transient operations are not faulty operations. However, it may affect building operational costs and sometimes, actions are needed to alter the control strategy (e.g., applying chiller sequencing controls to avoid high power demand).

Another top anomaly candidate is shown in Fig. 10. It corresponds to the daily operation on April 5th, 2015. It is observed that the actual building cooling load is constantly fluctuating, while the reconstructed data presents a relatively smooth trend. Further investigation shows that the root cause is improper chiller sequencing control logic, which leads to frequent on-off switches of chiller plants. Remedy measures should be taken to avoid unnecessary energy waste and reduce the risks in chiller operation faults.

5.2. Performance evaluation

Due to the unsupervised learning nature of autoencoders, it is typically not straightforward to assess the performance. As described in Section 3.4, the autoencoder performance is evaluated from two perspectives, i.e., the normality of reconstruction residuals and the quality of high-level features extracted by autoencoders.

5.2.1. Evaluation on reconstruction ability

A well-developed autoencoder has the ability to capture the intrinsic data behavior and therefore, the resulting reconstruction residuals can be regarded as random errors following a normal or near-normal distribution. As a visual example, Fig. 11 illustrates the reconstruction residual distribution of the feed-forward fully connected autoencoder without the use of conditional information. All the residual distributions under different masking noise levels present a bell shape with a mean of zero. The graphical results are similar for other autoencoders. The Shapiro-Wilk test is applied to examine the normality of reconstruction residuals of each sequence. A significant level of 5% is adopted for hypothesis testing. The percentages of residual sequences meeting the normality hypothesis are reported in Table 2. It is shown that the majority of residual sequences can pass the normality test, indicating that all autoencoders developed have solid reconstruction ability. In general, the percentages of residual sequences meeting the normality are higher for autoencoders with the 1D convolutional architecture. It indicates that 1D convolutional architecture is more suitable in analyzing time series data. The addition of *Month* as conditional information does not provide evident enhancement in terms of residual normality test results. Adding masking noises can occasionally increase the percentage of residual sequences passing the normality test. This method can be integrated with the process of autoencoder development, e.g., if the percentage of residual sequences passing the normality test is too low, further parameter tuning or model training is needed to avoid potential under-fitting problems.

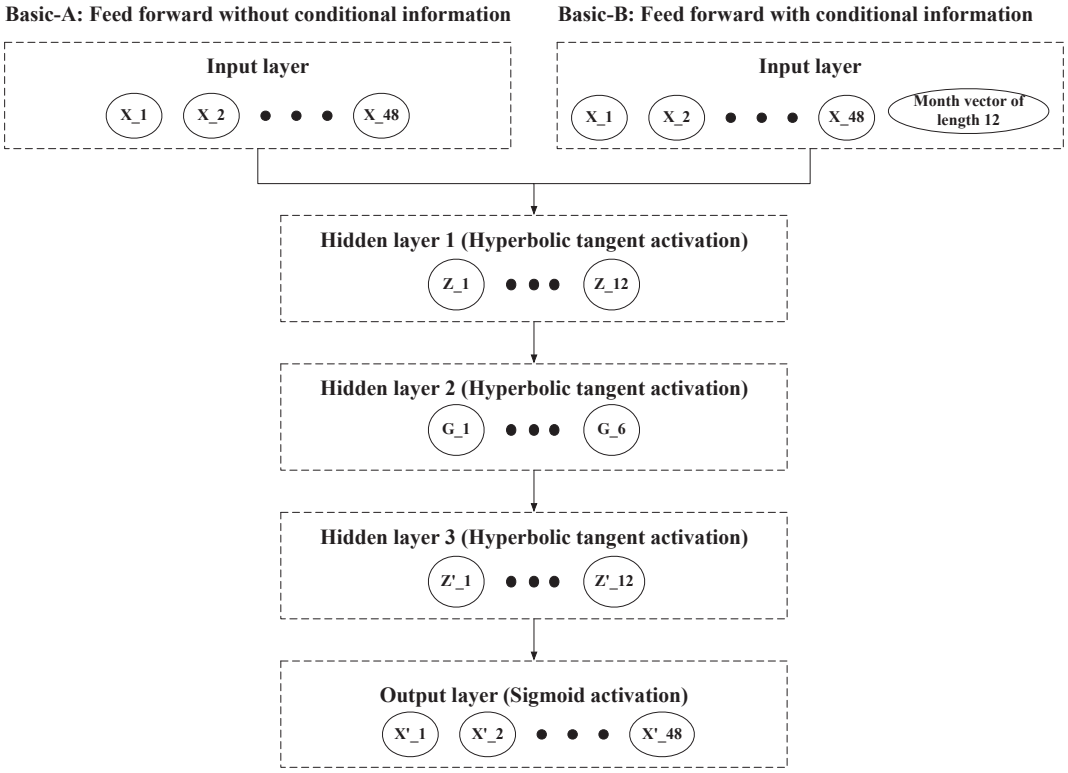

Fig. 6. Configurations of feed-forward autoencoders.

5.2.2. Evaluation on high-level feature quality

A good autoencoder should have the ability to extract meaningful and information-retaining features. Therefore, the autoencoder performance can be indirectly assessed based on the quality of high-level features extracted. In this study, a supplementary classification problem is designed to evaluate the usefulness of high-level features. As illustrated in Figs. 6 and 7, the number of high-level features for all autoencoders is fixed as six for comparison purposes. These high-level features are used as inputs to classify the *Day Type* of a subsequence. A more reliable autoencoder should produce more meaningful features and better reserve the information embedded in the original data. Hence, the resulting classification accuracy should be higher.

Considering the limited number of observations in testing data (i.e., the testing data consists of 355 complete daily subsequences), a simple neural network with two layers is used. It has one input layer with six neurons and one output layer with seven neurons (i.e., each neuron at the output layer represents a *Day Type*). The testing data are further divided into two sets according to proportions of 70% and 30% for model training and testing respectively. It should be noted that the relative performance rather than the absolute classification accuracy is the concern here. The classification results are shown in Fig. 12. The blue dotted horizontal line presents the classification accuracy using daily subsequences directly. The red dashed horizontal line presents the baseline accuracy using a naïve method, i.e., the most frequent *Day Type* in the testing data is used as prediction results. The error bar shows the 95% confidence interval on classification accuracy. The highest classification accuracy is achieved using the high-level features generated by the 1D convolutional autoencoder trained without conditional information and using a 10% masking noise level. In general, convolutional autoencoders have slightly better performance than the feed-forward fully connected autoencoders. The use of *Month* as conditional information does not necessarily contribute to the generation of more meaningful high-level features. By contrast, adding masking noises can improve the quality of high-level features occasionally. It is observed that when the masking noise level is relatively small, such as 5% or 10%, the high-level features extracted can be more useful for classification compared with those without masking noises. It may due to the existence of outliers or transient measurements in the original data. When the masking noise level is too high, a drop in the high-level feature quality can be observed, especially for convolutional autoencoders.

5.2.3. Evaluation on computation efficiency

The computation work associated with this research is performed using a MacBook Pro with a 2.5 GHz Intel Core i7 processor. The whole programming is done using the open-source software *R* [43].

Fig. 13 presents the computation time for each type of autoencoder. It is observed that the computation time is primarily affected by the autoencoder architecture and the feed-forward fully connected architecture results in much more efficient computation. Including the *Month* as conditional information does not lead to evident increase in the computation time of feed-forward fully connected autoencoders, while noticeable increase is observed in convolutional autoencoders. The increase in computation time is a result of the growth in total parameter number. The total parameter number of the basic feed-forward autoencoder is 1374. Since the conditional information of *Month* is integrated as a 12-value vector in feed-forward architecture, the total parameter number only increases slightly to 1518. By contrast, the conditional information needs to be integrated as matrices in 1D convolutional architecture and therefore, a dramatic increase in total parameter number from 763 to 1339 is observed. As a result, the growth in computation time of convolutional autoencoders is more noticeable. The change in masking noise levels is not a main affecting factor as it changes neither the dimension of input data nor the autoencoder configurations.

6. Conclusions

Accurate anomaly detection in building energy data can provide valuable clues for building professionals to improve building

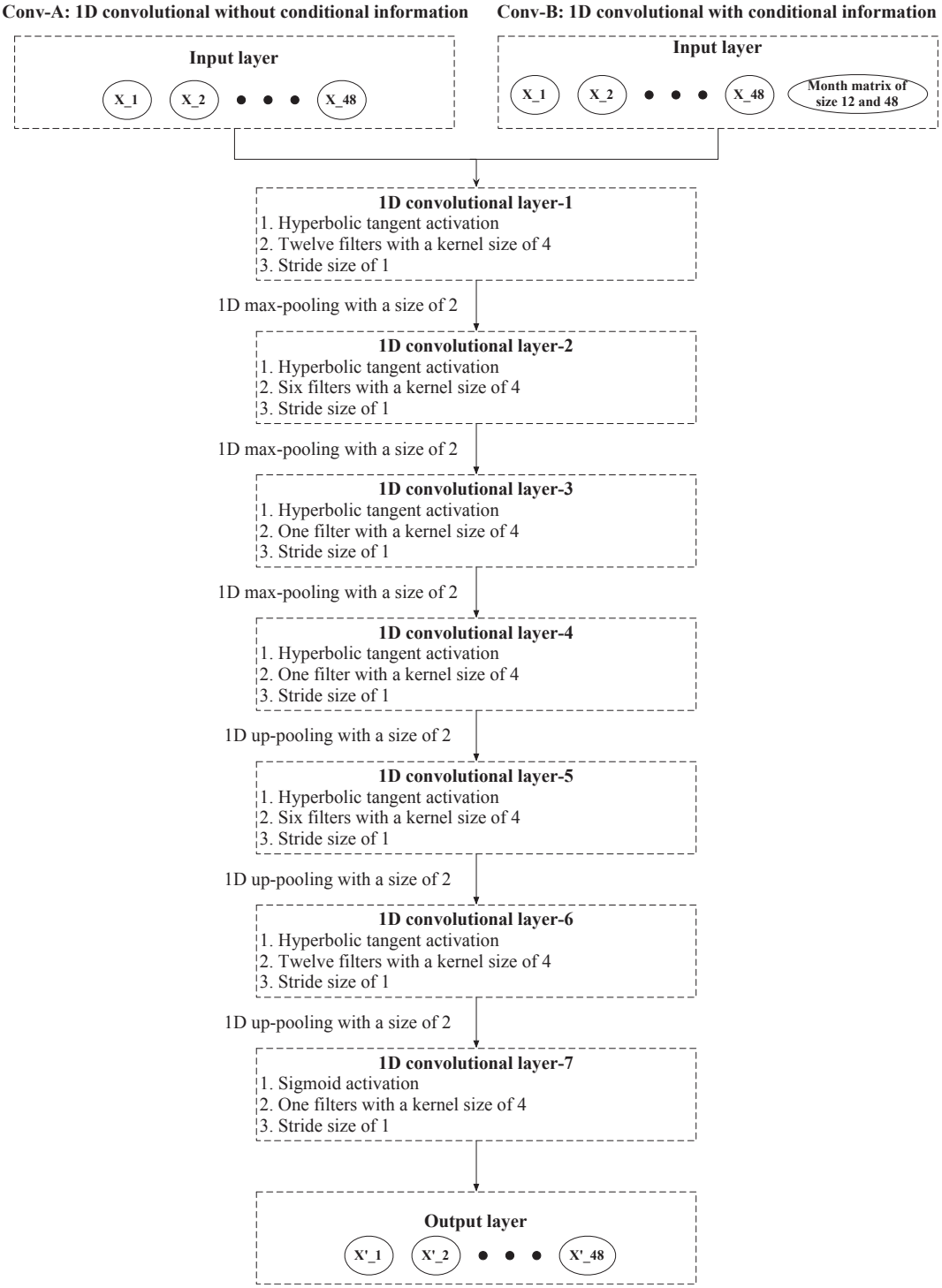

Fig. 7. Configurations of 1D convolutional autoencoders.

operational performance. Considering that the information on anomaly labels is usually limited or unknown, unsupervised anomaly detection is more promising for practical applications. Autoencoder is one of the most advanced techniques in unsupervised learning. It can be applied to extract high-level features from input sequence, based on which the expected or normal data behavior is projected. In such a case, autoencoders provide a dynamic performance benchmarking approach, i.e., the reconstructed sequence serves as the benchmark and is unique given different input sequences. The reconstruction losses can therefore be used as indicators for anomaly detection. This research investigates the usefulness of autoencoders in detecting anomalies in building energy data. The main contribution is summarized as follows: (1) an autoencoder-based ensemble method is proposed for the unsupervised anomaly detection. The ensemble is developed considering different autoencoder architectures and training schemes. To the best of the authors' knowledge, it is the first attempt in the building field. (2) Methods are proposed to indirectly evaluate the autoencoder performance. These methods are of high practical value when anomaly labels are unknown and directly evaluation on anomaly detection accuracy is infeasible.

The research results show that the proposed autoencoder-based ensemble method can successfully identify anomalies in building

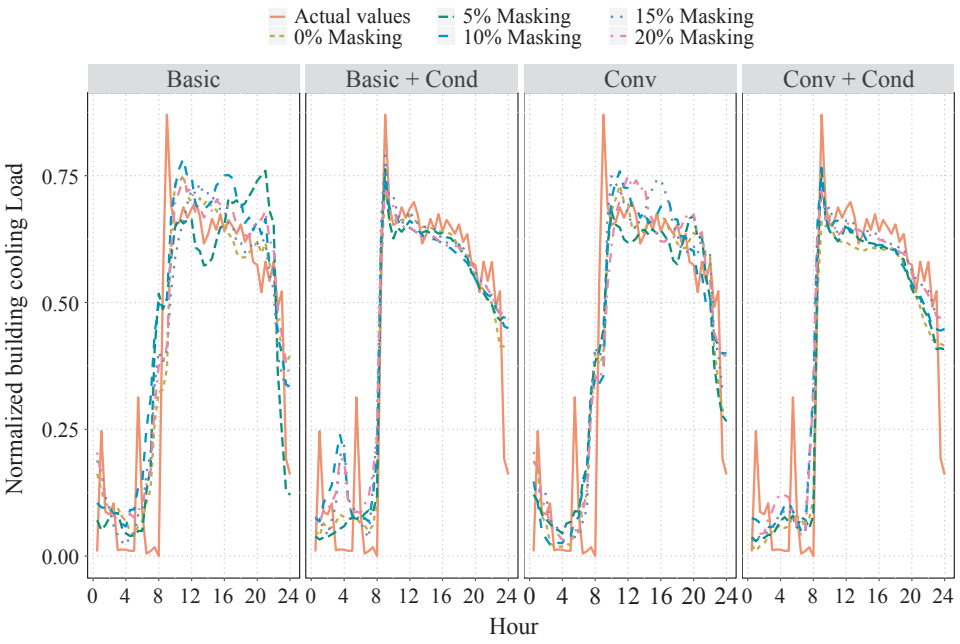

Fig. 8. An example of anomalies due to atypical operations.

energy data. Various types of anomalies have been discovered, including transient behaviors, operation faults, inefficient control strategies and atypical events. The insights obtained can help building professionals to better understand the building operating characteristics, based on which actionable measures can be developed for energy conservation and cost management. The method can be fully automated and integrated with Building Energy Management Systems for practical implementations. The final anomaly scores are easy to interpret. They range from zero to one and can be regarded as possibilities of being an anomaly.

The performance of different autoencoder architectures and training schemes has been evaluated and compared. The reconstruction residual distribution is used to inspect the autoencoder reconstruction ability. A normal or near normal distribution with a mean of zero should be observed if the autoencoder is well developed. This criterion can be integrated into the autoencoder training process to avoid the under-fitting problem. A supplementary classification task is formulated to indirectly evaluate the autoencoder reliability based on the usefulness of high-level features extracted for classification. In terms of the autoencoder architecture, it is shown that the 1D convolutional architecture can better preserve the information embedded in temporal data. The inclusion of *Month* as conditional information for autoencoder development leads to smoother reconstruction profiles. The reconstructed profiles can be used as better benchmarks taking into account the influence of seasonality. However, it does not have evident contribution to the quality of high-level features as shown by the supplementary classification results. It should be noted that the smoothing effect may be alleviated if other conditional information, such as *Day Type* and *Hour*, are introduced for autoencoder development. In addition, it is shown that adding a small proportion of masking noises (e.g., 5% or 10%) can enable autoencoders to learn more reliable and robust features from real-world data. It is especially useful for analyzing building

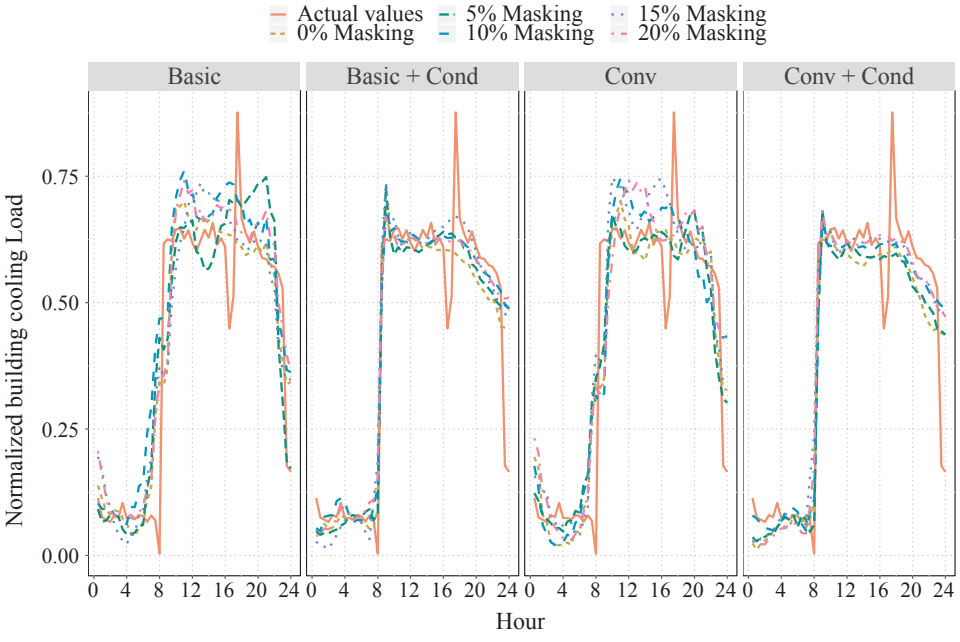

Fig. 9. An example of anomalies due to transient operations.

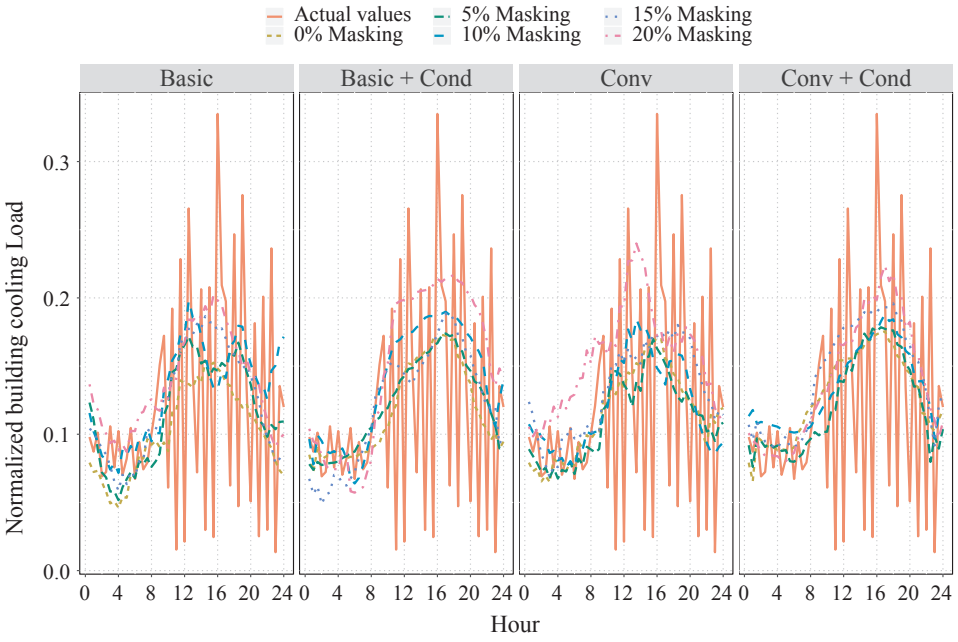

Fig. 10. An example of anomalies due to improper control strategies.

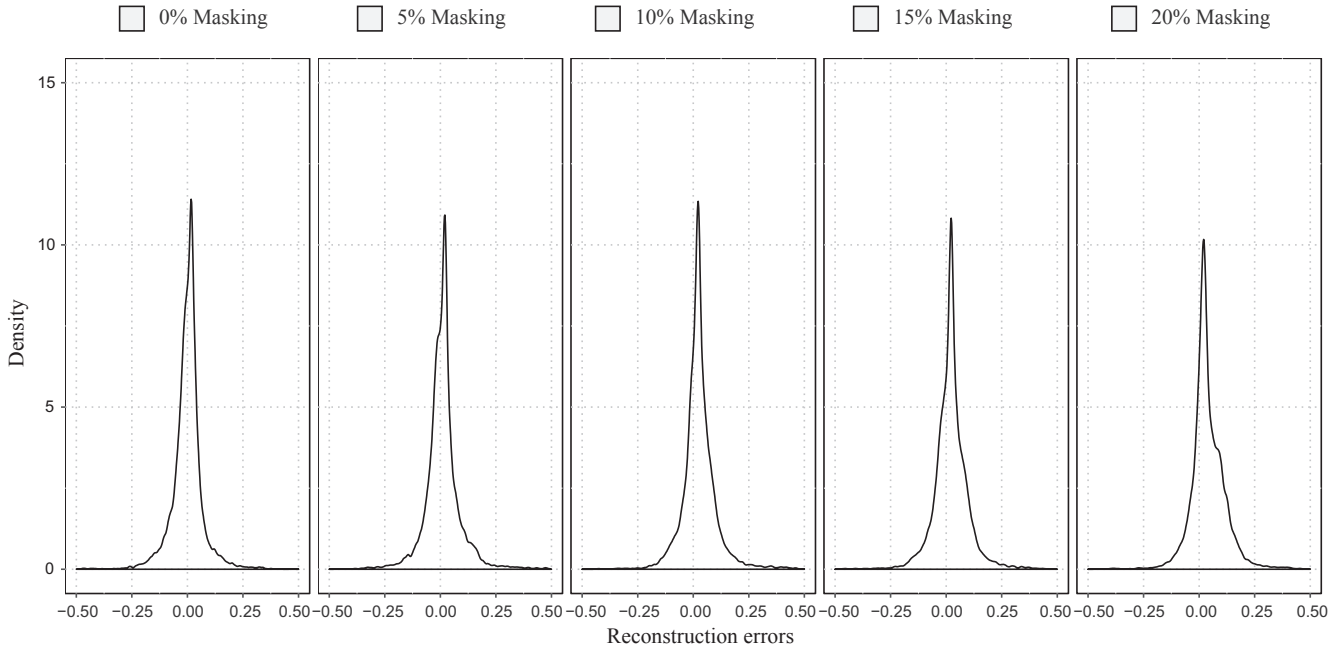

Fig. 11. Reconstruction residual distributions of the *Basic-A* autoencoders.

Table 2
Percentages of residual sequences passing the normality test (significant level = 0.05).

Autoencoders	Basic-A	Basic-B	Conv-A	Conv-B
0% masking	0.88	0.84	0.90	0.90
5% masking	0.83	0.86	0.91	0.93
10% masking	0.84	0.86	0.91	0.88
15% masking	0.84	0.86	0.90	0.90
20% masking	0.79	0.83	0.91	0.92

operational data with large numbers of missing values, outliers and transient measurements. Previous studies mainly perform data analysis based on the original data format and therefore, complicated data preprocessing is needed to enhance data quality, e.g., applying predictive models to fill in missing values. The results obtained in this study indicate that an alternative approach to data analysis is to use the high-level features extracted by denoising autoencoders. It can greatly alleviate the data preprocessing workload (e.g., missing values transformed as zeros) without negatively affecting the quality of data analysis results. Further studies will focus on exploring the potential of other autoencoder architectures (e.g., recurrent autoencoder architecture) and conditional variables (e.g., *Day Type*, *Hour* and chiller power consumption) in unsupervised anomaly detection.

Acknowledgements

The authors gratefully acknowledge the support of this research by the National Nature Science Foundation of China (Grant Nos. 71772125 and 51706197) and the Natural Science Foundation of SZU (Grant No. 2017061).

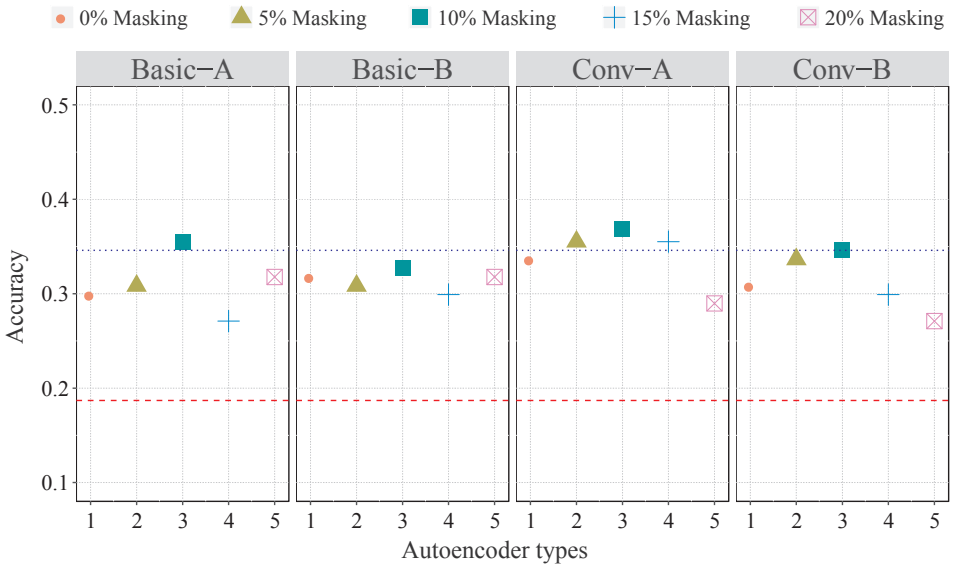

Fig. 12. Supplementary classification results.

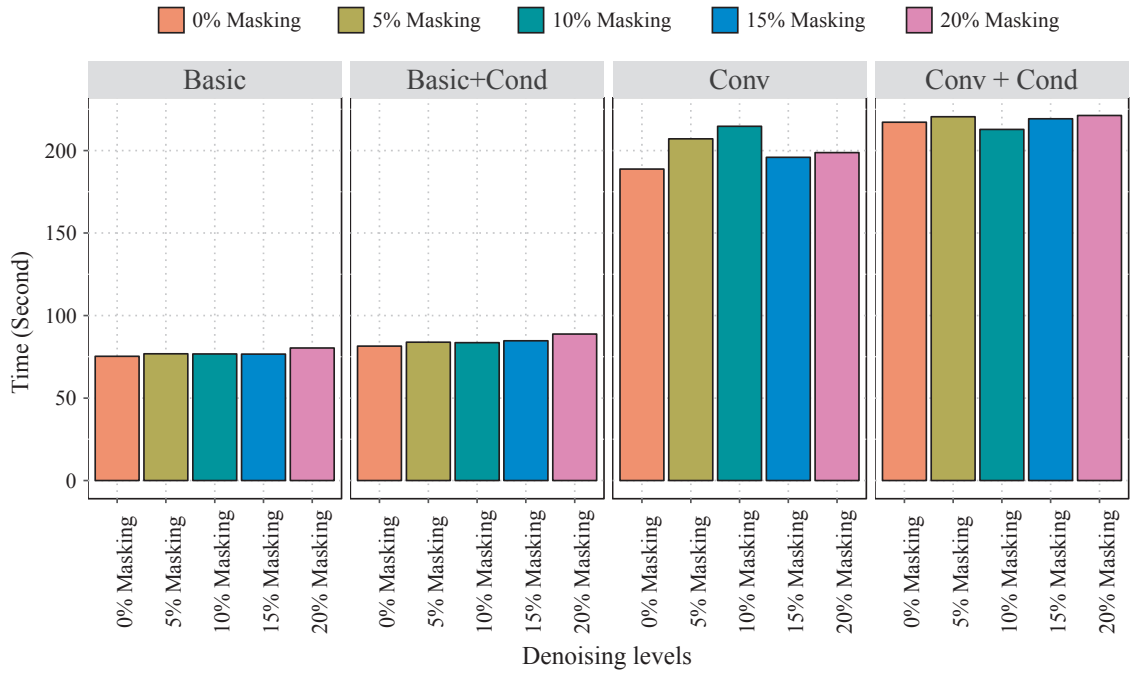

Fig. 13. Computation time for autoencoder development.

References

[1] Urge-Vorsatz D, Cabeza LF, Serrano S, Barreneche C, Petrichenko K. Heating and cooling energy trends and drivers in buildings. Renew Syst Energy Rev 2015;41:85–98.

[2] Lee D, Cheng CC. Energy savings by energy management systems: a review. Renew Sust Energy Rev 2016;56:760–77.

[3] Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv 2009;41:1–58.

[4] Goldstein M, Uchida S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS ONE 2016;11:e0152173.

[5] Araya DB, Grolinger K, El Yamany HF, Capretz MAM, Bitsuamlak G. An ensemble learning framework for anomaly detection in building energy consumption. Energy Build 2017;144:191–206.

[6] Wang HL, Xu P, Lu X, Yuan DK. Methodology of comprehensive building energy performance diagnosis for large commercial buildings at multiple levels. Appl Energy 2016;169:14–27.

[7] Zhao HX, Magoules F. A review on the prediction of building energy consumption. Renew Sust Energy Rev 2012;16:3586–92.

[8] Du ZM, Fan B, Jin XQ, Chi JL. Fault detection and diagnosis for buildings and HVAC systems using combined neural networks and subtractive clustering analysis. Build Environ 2014;73:1–11.

[9] Magoules F, Zhao HX, Elizondo D. Development of an RDP neural network for building energy consumption fault detection and diagnosis. Energy Build 2013;62:133–8.

[10] Zhao Y, Wang SW, Xiao F. Pattern recognition-based chiller fault detection method using support vector description (SVDD). Appl Energy 2013;112:1041–8.

[11] Guo YB, Li GN, Chen HX, Wang JY, Guo MR, Sun SB, Hu WJ. Optimized neural network-based fault diagnosis strategy for VRF system in heating mode using data mining. Appl Therm Eng 2017;125:1402–13.

[12] Seem JE. Pattern recognition algorithm for determining days of the week with similar energy consumption profiles. Energy Build 2005;37:127–39.

[13] Seem JE. Using intelligent data analysis to detect abnormal energy consumption in buildings. Energy Build 2007;39:52–8.

[14] Li XL, Bowers CP, Schnier T. Classification of energy consumption in buildings with outlier detection. IEEE Trans Ind Electron 2010;57:3639–44.

[15] Li MC, Miao L, Shi J. Analyzing heating equipment's operations based on measured data. Energy Build 2014;82:47–56.

[16] Wang SW, Cui JT. Sensor-fault detection, diagnosis and estimation for centrifugal chiller systems using principal-component analysis method. Appl Energy 2005;82:197–203.

[17] Ma ZJ, Yan R, Nord N. A variation focused cluster analysis strategy to identify typical daily heating load profiles of higher educational buildings. Energy

2017;134:90–102.
[18] Molina-Solana M, Ros M, Ruiz MD, Gomez-Romero J, Martin-Bautista MJ. Data science for building energy management: a review. Renew Sust Energy Rev 2017;70:598–609.
[19] Capozzoli A, Lauro F, Khan I. Fault detection analysis using data mining techniques for a cluster of smart office buildings. Expert Syst Appl 2015;42:4324–38.
[20] Yu Z, Haghighat F, Fung CM, Zhou L. A novel methodology for knowledge discovery through mining associations between building operational data. Energy Build 2012;47:430–40.
[21] Xue PN, Zhou ZG, Fang XM, Chen X, Liu L, Liu YW, Liu J. Fault detection and operation optimization in district heating substations based on data mining techniques. Appl Energy 2017;205:926–40.
[22] Cabrera DFM, Zareipour H. Data association mining for identifying lighting energy waste patterns in educational institutes. Energy Build 2013;62:210–6.
[23] Fan C, Xiao F, Yan CC. A framework for knowledge discovery in massive building automation data and its application in building diagnostics. Automat Constr 2015;50:81–90.
[24] Li GN, Hu YP, Chen HX, Li HR, Hu M, Guo YB, et al. Data partitioning and association rule mining for identifying VRF energy consumption patterns under various part loads and refrigerant charge conditions. Appl Energy 2017;185:846–61.
[25] Fan C, Xiao F, Madsen H, Wang D. Temporal knowledge discovery in big BAS data for building energy management. Energy Build 2015;109:75–89.
[26] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;321:436–44.
[27] Langkvist M, Karlsson L, Loutfi A. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognit Lett 2014;41:11–24.
[28] Baldi P. Autoencoders, unsupervised learning, and deep architectures. JMLR Workshop Conf Proc 2012;27:37–50.
[29] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
[30] Masci J, Meier U, Ciresan D, Schmidhuber J. Stacked convolutional auto-encoders for hierarchical feature extraction. In: Proc of the 21st ICANN 2011, Part I; 2011. p. 52–9.
[31] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural network overfitting. J Mach Learn Res 2014;15:1929–58.
[32] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. Proc Adv Neural Inform Process Syst 2012;25:1090–8.
[33] Lipton ZC, Berkowitz. A critical review of recurrent neural networks for sequence learning; 2015. Available from: arXiv:1506.00019.
[34] Miller C, Nagy Z, Schlueter A. Automated daily pattern filtering of measured building performance data. Automat Constr 2015;49:1–17.
[35] Proakis JG, Manolakis DG. Digital signal processing. Prentice-Hall; 2000.
[36] Dietterich TG. Ensemble methods in machine learning. Lecture notes in computer science. Calgari (Italy); 2000.
[37] Zimek A, Gaudet M, Campello RJB, Sander J. Subsampling for efficient and effective unsupervised outlier detection ensembles. In: Proc of the 19th ACM SIGKDD; 2013. p. 428–36.
[38] Lazarevic Z, Kumar V. Feature bagging for outlier detection. In: Proc of the 11th ACM SIGKDD; 2005. p. 157–66.
[39] NIST/SEMATECH. E-Handbook of statistical methods. U.S. Department of Commerce; 2012. < http://www/itl.nist.gov/div898/handbook/ > .
[40] Thode HJ. Testing for normality. New York: Marcel Dekker; 2002.
[41] Ghasemi A, Zahediasl S. Normality tests for statistical analysis: a guide for non-statisticians. Int J Endocrinol Metab 2012;10:486–9.
[42] Jayalakshmi T, Santhakumaran A. Statistical normalization and back propagation for classification. Int J Comput Theor Eng 2011;3:89–93.
[43] R Development Core Team. R: a language and environment for statistical computing. Vienna (Austria): R Foundation for Statistical Computing; ISBN 3-900051-07-0; 2008. < http://www.R-project.org > .