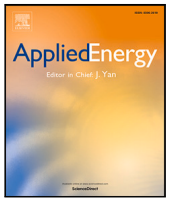
Applied Energy 334 (2023) 120701

Contents lists available at ScienceDirect

Applied Energy

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



Augmenting energy time-series for data-efficient imputation of missing values

Antonio Liguori [a,*], Romana Markovic [b], Martina Ferrando [c], Jérôme Frisch [a], Francesco Causone [c], Christoph van Treeck [a]

[a] E3D - Institute of Energy Efficiency and Sustainable Building, RWTH Aachen University, Mathieustr. 30, 52074 Aachen, Germany
[b] Building Science Group, Karlsruhe Institute of Technology, Englerstr. 7, 76131 Karlsruhe, Germany
[c] Department of Energy, Politecnico di Milano, Via Lambruschini 4, 20156 Milano, Italy

ARTICLE INFO

Keywords:
Missing data
Data augmentation
Data scarcity
Building energy data
Deep learning

ABSTRACT

This study explores the applicability of data augmentation techniques for reconstructing missing energy time-series in limited data regimes. In particular, multiple synthetic copies of a relatively small training dataset are stacked together with pseudo-random noise. First, an existing convolutional denoising autoencoder is selected from a previous work, as the base imputation model of this study. Then, an optimal augmentation rate, which minimizes the training set of the model, is chosen based on the preliminary results obtained from one building. The results proved that, augmenting 80 times a nine days-long training set could reduce the initial average root mean squared error (RMSE) by 37% and 48%, for continuous and random missing scenarios. Additionally, the augmented model outperformed the benchmark methods with 23% and 12% lower average RMSE. No additional tuning or calibration costs were required for the existing base imputation model. Therefore, the presented data augmentation technique could significantly reduce the expensive computational costs associated with deep learning models.

1. Introduction

Buildings play a major role in the worldwide consumption of energy resources [1]. For that purpose, building energy data estimation and forecasting is of primary importance for reducing the global CO_2 emissions produced by buildings' operation [2]. In recent years, the increasing use of data-driven methods has led to remarkable advances in the energy-related building research. Among these methods, machine learning models have been widely applied [3].

Machine learning refers to the ability of a system to extract useful knowledge from raw data [4]. Therefore, the performance of a machine learning model might be significantly affected when relevant information is missing in a dataset. This is a typical problem for building practitioners, as frequent faults during the data collection and transmission often cause missing values in the datasets [5]. Common approaches to replace these faulty data points are the mean or zero imputation, for example in the work of Chong et al. [6]. However, the use of these simplified methods can drastically change the statistical distribution of the original dataset, hindering energy analysis [7].

With the advances of deep learning, different advanced solutions have been proposed to impute missing values in data time-series. In the literature, these methods have often been able to outperform more simplified statistical and machine learning-based techniques [8–22] (see Section 2). However, few studies have been found to deal with the extreme working conditions of building applications, such as data scarcity. Namely, not every building might have sufficient availability of historical data, for example buildings at the initial stage of operation [23]. This is a particular problem when applying deep learning-based imputation models, as they usually require an extensive amount of training data [5]. Common approaches to reduce the computational requirements of deep learning models are changing the network architecture or hyperparameters [24,25]. For example, Silka et al. [24] adopted the learning rate decay technique and explored different optimization algorithms for a Long Short Term Memory-Recurrent Neural Network (LSTM-RNN)-based deep learning model. Kreuzer et al. [25] investigated the activations in the convolutional layers of a 2D Convolutional-LSTM network and eliminated the irrelevant channels for prediction. Based on these considerations, most of the models proposed in the literature require specific architectures, e.g. hyperparameters, on each dataset. Despite the recent advances in automatic machine learning (AutoML), architecture optimization still largely depends on human expertise and requires a significant amount

* Corresponding author.
E-mail address: liguori@e3d.rwth-aachen.de (A. Liguori).

https://doi.org/10.1016/j.apenergy.2023.120701
Received 5 July 2022; Received in revised form 3 December 2022; Accepted 14 January 2023
Available online 23 January 2023

of computational resources [26]. Therefore, from a building practitioner's point of view, the field application of the state-of-the-art deep learning-based imputation models might be particularly challenging.

In order to facilitate the integration of these methods with building automation systems (BAS), standardization and documentation are extremely important [27]. According to the international guidelines [28], missing values in BAS might be considered "non annotated faults" and handled with the methods for fault detection and diagnosis (FDD). When dealing with time-series that have clear periodic profile patterns, such as daily electric load patterns [29], these methods are usually trained to identify atypical behaviors in the observed data [30]. In the literature, the models for FDD of building automation systems have been successfully researched with a methodological focus on the autoencoder neural networks [30–33]. Autoencoders turn the input data into a compressed representation by the encoder, while the decoder learns to reconstruct the original data starting from the last encoded state [4]. However, based on previous research [34], the massive amount of training data and computational resources required for optimization limit the field application of these models.

1.1. Contribution

Based on the previous considerations, it is observed that a successful imputation model for building energy data time-series should satisfy the following requirements. (1) It should be complex enough to reflect the non-linear characteristics of the faulted energy data. (2) It should not require excessive human expertise or computational resources for architecture optimization. (3) It should not require extensive historical data for training.

While autoencoder neural networks can generally satisfy the first criteria [30], they often require additional optimization and a massive amount of training data [34]. In the literature, the first solution to this problem has been the application of transfer learning [35]. However, the use of transfer learning is not always warranted due to the differences in the latent space of each domain. Namely, the time-series in the source and target domain should be sufficiently similar to avoid negative transfer [36]. As highlighted by Pinto et al. [36], there are still no recognized guidelines on how to avoid negative transfer for BAS.

To this end, this paper is proposing a simple, yet effective, methodology to impute missing energy time-series with existing denoising autoencoder neural networks. This methodology relies on a particular data augmentation technique (Aug) which consists in stacking together multiple synthetic copies of a relatively small training dataset with pseudo-random noise (see Section 3.1). The presented solution can be easily adopted by building practitioners, as it is widely applicable to different settings. In order to prove the previous point, experiments are performed with an existing denoising convolutional autoencoder (CAE) which proved to be effective in a completely different setting, i.e. missing indoor environmental quality (IEQ) data time-series imputation [34]. Differently from the original model, namely CAE, the analyzed autoencoder-augmentation framework (CAE + Aug) outperforms the used benchmark methods with 23% and 12% lower average RMSE. Therefore, the results suggest that the proposed data augmentation technique can boost the performance of the same denoising autoencoder on generic time-series that share similar daily periodicity and variability.

Table 1 defines the abbreviations introduced in this paper. The rest of the work is organized as follows: Section 2 presents the related recent advances in the literature. Section 3 provides the theoretical foundations for the concept of denoising. Section 4 introduces the datasets used in this paper. Section 5 presents the adopted methodology. Section 6 investigates the final results for continuous and random missing scenarios. The novel findings are discussed and summarized in Sections 7 and 8.

2. Literature review

2.1. Recent advances in deep learning-based imputation of missing data time-series

In the literature, deep learning-based imputation methods have been extensively explored to handle missing values in data time-series [8–22]. In particular, different papers have been found to deal with missing energy data [8–14]. Ma et al. [8] proposed a hybrid Long Short Term Memory model with Bi-directional Imputation and Transfer Learning (LSTM-BIT) to impute missing electric consumption data from a university lab building. Here, the base model was pre-trained on data from a similar building while transfer learning was used to avoid network saturation issues. In particular, the network architecture and hyperparameters were initialized on the source data, while transfer learning was used to fine-tune them on the target data. The corrupted dataset was used for evaluation, while the rest was for training, i.e. minimum 10%. The results proved that the proposed framework could effectively deal with different missing data scenarios, such as continuous and random missing. However, the proposed method relied on the prior assumption that source and target data collected on sufficiently similar buildings were available. Hence, the generalization capabilities to different settings were not sufficiently explored. Li et al. [9] presented a back propagation neural network (BPNN) to successfully impute missing air-conditioning power consumption data from a public building. The model was trained on one whole month of data, consisting of multiple indoor and outdoor parameters. The hyperparameters were also optimized based on empirical formulations. Liu et al. [10] developed a complex model based on a sparse autoencoder with a coordinate descendant optimization algorithm (SAE-CD). They applied this method for handling missing data from supervisory control and data acquisition (SCADA) systems of 15 wind turbines In total, two-month-long datasets were available, 40% of which were used for training. Additionally, an extensive hyperparameter search was necessary to achieve good results. Experiments with other machine learning-based imputation approaches demonstrated the superiority of this method. Hussain et al. [11] implemented an hybrid convolutional neural network and long-short term memory model (CNN-LSTM) to impute missing values in air-conditioning appliances time-series datasets. The hyperparameter tuning was essential to optimize the aforementioned framework. Additionally, 67.7% of data was used for training. Compared to the single CNN and LSTM variants, the hybrid approach had higher performance. Jung et al. [12] proposed a deep learning-based missing value imputation method based on an ensemble of multi layer perceptrons (MLPs). The proposed framework was based on a softmax ensemble network (SENet). In particular, 26 different variables were used as input features, including indoor, weather and calendar data. The corrupted dataset was used for evaluation, while the rest for training, i.e. minimum 70%. The output variable was the missing electricity consumption data. Even in this case, extensive hyperparameter tuning was necessary to achieve accurate results. The model outperformed different advanced imputation methods, such as random forest (RF), support vector machine (SVM), XGBoost and other deep learning-based methods. Ma et al. [13] explored linear memory vector recurrent neural network (LIME-RNN) for imputing missing values in different time-series, including electricity consumption data. Even if a same model architecture was used for different data streams, the model still relied on an excessive number of training data, i.e. 70% of the total dataset. The algorithm achieved state-of-the-art performance compared to other statistical and machine learning based imputation methods. Finally, Zerveas et al. [14] implemented a transformer-based framework (TNN) for handling faulted measurements in several real-world time-series from various domains, including energy appliances. The optimal model's architecture was tuned separately for each dataset. On average, the model was trained on 49% of the available datasets, outperforming other state-of-the-art supervised methods.

Table 1
List of abbreviations.

Abbreviation	Definition	Abbreviation	Definition
AENN	autoencoder neural networks	LI	linear interpolation
Aug	augmentation rate	LIME	linear memory vector
AutoML	automatic machine learning	LSTM	long short-term memory
BAS	building automation system	MAE	mean absolute error s
BIT	bi-directional imputation and transfer learning	MFA	mixture factor analysis
BPNN	back propagation neural network	MI	mean interpolation
CAE	convolutional denoising autoencoder	MLP	multi layer perceptron
CD	coordinate descendant	MuSDRI	multi-seasonal decomposition based recurrents
CNN	convolutional neural network	NRMSE	normalized root mean squared error
CR	corruption rate	OB	occupant behavior
CVAE	conditional variational autoencoders	RF	random forest
DL	deep learning	RH	indoor relative humidity
DNN	deep neural networks	RMSE	root mean squared error
EMD	empirical mode decomposition	RNN	recurrent neural network
FDD	fault detection and diagnosis	SA	self attention
GAIN	generative adversarial imputation network	SAE	sparse autoencoder
GAN	generative adversarial network	SCADA	supervisory control and data acquisition
GAN	generative adversarial network	SENet	softmax ensemble network
GRU	gated recurrent unit	SSIM	self-attention generative adversarial imputation net
IEQ	indoor environmental quality	STNN	spatio-temporal neural network
IQR	interquartile range	SVM	support vector machine
KDE	kernel density estimation	T	indoor air temperature
KNN	k-nearest neighbor	TNN	transformer-based framework

Table 2
Comparison of the proposed modeling approach with deep learning-based imputation methods for time-series. Please refer to Table 1 for the abbreviations. "Modeling approach" refers to the proposed deep learning-based imputation model, "Domain of application" is the specific domain of the analyzed time-series. "Arch. optimization" provides information on whether additional tuning was required, "Training data" indicates the ratio between the training and total data.

Study	Modeling approach	Domain of application	Arch. Optimization	Training data [0–1]
Ma et al. [8]	LSTM-BIT	Energy	Partial	0.10
Li et al. [9]	BPNN	Energy	Yes	0.96
Liu et al. [10]	SAE-CD	Energy	Yes	0.40
Hussain et al. [11]	CNN-LSTM	Energy	Yes	0.67
Jung et al. [12]	MLP + SENet	Energy	Yes	0.70
Ma et al. [13]	LIME-RNN	Energy and others	No	0.70
Zerveas et al. [14]	TNN	Energy and others	Yes	0.49
Delasalles et al. [15]	STNN	Healthcare and others	Yes	0.38
Li et al. [16]	EMD-LSTM	Structural dynamics	Yes	0.83
Zhang et al. [17]	SSIM	Hydrology	Yes	0.87
Flores et al. [18]	GRU + Aug	Meteorology	Yes	0.66
Chen et al. [19]	EMD-LSTM	Structural dynamics	Yes	0.80
Flores et al. [20]	LSTM	Meteorology	Yes	0.97
Zhou et al. [21]	MuSDRI	Hydrology and others	Yes	0.75
Zhang et al. [22]	SA-GAIN	Traffic flow	Yes	0.80
	CAE + Aug	Energy	No	0.025

In summary, most of the papers that have present deep learning-based imputation models for missing energy data report state-of-the-art performance. However, most of these models require excessive human expertise or computational resources for architecture optimization. Furthermore, they also require extensive historical data for training. Therefore, the field application of these models might be particularly challenging. Since energy data are basically time-series, the literature review of this Section is further extended to different domains of application. Table 2 highlights the strengths of the proposed modeling approach of this paper, i.e. CAE + Aug, compared to each of the relevant imputation methods found in the literature. In particular, each study was reviewed in terms of modeling approach, domain of application, architecture optimization and training data. Here, "modeling approach" refers to the proposed deep learning-based imputation model, while "domain of application" is the specific domain of the analyzed time-series, e.g. meteorology. Furthermore, "architecture optimization" provides information on whether additional tuning was required, while "training data" indicates the ratio between the training and total data (an average value is provided for multiple time-series). In particular, only the studies that clearly report all the previous points are shown in the Table.

As shown in Table 2, compared to the deep learning-based imputation methods proposed in the literature, CAE + Aug requires neither additional architecture optimization nor extensive historical data. Additionally, differently from transfer learning-based methods, it does not require pre-training or optimization on data collected from similar settings. This is possible thanks to the defined data augmentation strategy, which is explicitly designed to take advantage of the training, validation and evaluation criterion of denoising autoencoder neural networks (see Section 3).

2.2. Recent advances in data time-series augmentation

As described by Goodfellow et al. [4], data augmentation is the process of increasing a dataset with fake data, in order to improve model generalization. In literature, some studies have been recently published related to data augmentation techniques for building energy data. However, these works have often relied on computational expensive approaches based on generative models or building performance simulation [37–39]. Fan el al. [37] proposed a novel generative modeling-based data augmentation process for building energy data. Here, Conditional Variational Autoencoders (CVAE) were used to generate synthetic data samples, so that to train a predictive model based on Artificial Neural Networks. The results proved that variational autoencoder could be used to generate high quality data samples in order to boost the performance of predictive deep learning models.

The data augmentation process based on noise injection was instead discarded due to poor performance. Similarly to Fan et al. [37], Wu et al. [38] implemented a generative modeling-based data augmentation process for building energy data. However, here the generation of synthetic data was performed by a Generative Adversarial Network (GAN). Additionally, an ensemble method consisting of five different models was trained on the generated dataset in order to forecast the energy consumption of large commercial buildings. The described augmentation process decreased the original prediction error by 4.72%. Lu et al. [39] presented a data augmentation strategy based on the generation of synthetic data through calibrated building performance simulation and transfer learning. First, a LSTM model was pre-trained on the augmented dataset. Finally, a transfer learning technique was applied to fine-tune the pre-trained weights of the network on a limited data sample collected in the target building. It was shown that the augmentation process decreased the short-term predictive error of the deep learning model by 18.14%.

In summary, data augmentation for building energy data modeling often relies on generative approaches or building performance simulation. More simplified approaches with no human expertise, such as random noise injection, are often discarded due to poor performance. However, to the best of the authors' knowledge, these latter techniques have been mostly applied as a solely regularization strategy for deep learning models. As explained in the following Section, the use of noise injection as data augmentation strategy might considerably boost the performance of a denoising autoencoder neural network applied to missing data imputation problems. This consideration is further enriched by empirical assessments in Section 6.

3. Theoretical background

Machine learning models can be used to perform various tasks, such as classification or regression. Here, we define a task as the way to process a sample x with length n (i.e. with n components or features), measured from a particular object or event [4]. As such, by targeting different kind of tasks, a machine learning algorithm can let a computer learn from different kind of experiences. The final performance depends on the learning direction followed by the model. In particular, in the scope of this paper, the imputation of missing values is considered to be the result of a more generic task, called denoising.

3.1. Denoising

Denoising is defined as the task of cleaning or restoring a corrupted sample x^*, where i components of x are missing or anomalies due to some added noise v [4]. The corrupted sample is therefore defined as follows [40]:

$$x^* = x + v \ , \tag{1}$$

where the aim of the denoising task is to make x^* as close as possible to x, by reducing v. As shown in Vincent et al. [41], Eq. (1) is the result of a stochastic mapping $q(x^*|x)$ which can be realized by different corruption processes, such as Gaussian, masking and salt-and-pepper noise.

Gaussian noise consists in adding to the original samples statistical noise following a Gaussian distribution, as follows [42]:

$$v_{gaussian}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{1}{2}(x-\mu)^2/\sigma^2) \ , \tag{2}$$

where μ and σ are the mean and standard deviation of the gaussian distribution. As such, different levels of noise can be added to each sample, by changing the magnitude of the standard deviation [37]. By introducing small variations in x, Gaussian noise can be used to simulate malfunctions in the building sensors which lead to anomalies in the dataset.

Masking noise takes a fixed number d of random components for each sample x and imputes their original values with zero [41]. The denoising task involves the complete reconstruction of the missing information and it is, therefore, strictly correlated with the imputation of missing values. In particular, this kind of noise is used in this paper to simulate two different scenarios often occurring in smart building meter datasets, namely random and continuous missing. Random missing imputation is a relatively simpler task than continuous missing imputation, as the non corrupted entries are usually closer to the corrupted ones [8]. Given a fixed corruption rate $CR \in [0, 1]$, the number of corrupted components for each sample is given as follows:

$$d = n \cdot CR \ . \tag{3}$$

For the random missing scenario, the probability of a component i to be selected and forced to zero is drawn from a discrete uniform distribution [43]:

$$p(i) = 1/n \ . \tag{4}$$

Hence, the random missing masking noise v_{random} can be rewritten using the pseudo-code Algorithm 1.

Algorithm 1 Random missing masking noise

Require: An array x and a corruption rate CR
Ensure: Random noise to add to Equation 1
  function $v_{random}(x, CR)$
    $n \leftarrow$ length of x
    $d \leftarrow n \cdot CR$
    $x^1 \leftarrow x$
    $indices \leftarrow d$ random integers in $[0, n]$ ▷ $indices$ is an array of integers
    for i from 0 to n **do**
      if i is in $indices$ **then**
        $x^1[i] \leftarrow -x[i]$
      else if i is not in $indices$ **then**
        $x^1[i] \leftarrow 0$
      end if
    end for
    return x^1
  end function

By introducing the masking noise v_{random} in Eq. (1), a corrupted sample x^* with randomly missing components is obtained. This approach is also followed to compute the continuous missing random noise $v_{continuous}$. Here, d defines both the number of components set to zero and the length of the missing continuous interval. In particular, the first element of the interval is randomly selected following the distribution of Eq. (4) and the remaining corrupted components are selected with a forward expansion of size d. The resulting algorithm can be observed in the pseudo-code Algorithm 2.

By introducing the masking noise $v_{continuous}$ in Eq. (1), a corrupted sample x^* with continuous missing components is obtained.

Salt-and-pepper noise can be considered as a special case of masking noise, where the selected components are set either to their minimum or maximum value [4]. As this approach is recommended for binary problems, it is not followed in this paper.

3.2. Denoising as a training, validation and evaluation criterion

As described in Section 2, in the context of building energy data modeling, the use of noise injection as data augmentation strategy has been mostly applied as a solely regularization strategy for deep learning models. In the generic machine learning research, regularization is defined as "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error" [4]. As such, it is used to avoid overfitting issues in either too simple or too complex machine learning models [44]. In particular,

Algorithm 2 Continuous missing masking noise

Require: An array x and a corruption rate CR
Ensure: Continuous noise to add to Equation 1
  function $v_{continuous}(x, CR)$
    $n \leftarrow$ length of x
    $d \leftarrow n \cdot CR$
    $x^1 \leftarrow x$
    index $\leftarrow$ random integer in $[0, n]$ ▷ $index$ is an integer
    while $index + d >= n$ **do**
      index $\leftarrow$ random integer in $[0, n]$
    end while
    for i from 0 to n **do**
      if $i >= index$ and $i <= index + d$ **then**
        $x^1[i] \leftarrow -x[i]$
      else if $i < index$ or $i > index + d$ **then**
        $x^1[i] \leftarrow 0$
      end if
    end for
    return x^1
  end function

regularization can be applied by introducing a penalty constraint ϵ to an error function E, as follows [44]:

$$E^* = E + \epsilon \,, \tag{5}$$

where E^* is the final error function that the learning model has to minimize. As explained by Bishop et al. [44], adding small noise to the input of a neural network has the same effect of adding a penalty constraint to its error function. Therefore, for small values of σ, applying Eq. (5) is equivalent to corrupting an original sample x with Gaussian noise. The same consideration is not exactly true for masking or salt-and-pepper noise, since they both drastically corrupt the original data [4].

Fan et al. [37] showed how Gaussian noise could be used as data augmentation strategy for predictive deep learning models, by introducing small changes in the input data. As explained by the same authors, this would potentially make a predictive model more robust to malfunctions in the building sensors (i.e. anomalies in the data), by boosting its performance on these conditions. However, when compared to more complex data augmentation strategies, such as generative approaches based on CVAE, the former method could not satisfactorily increase the accuracy of the predictive model. The reasons behind the previous results might be diverse. Among them, it should be noticed that there was no strong correlation between the task and the data augmentation strategy. As observed in the paper, different copies of a same dataset were created with different Gaussian noise for training, but the testing was, indeed, performed on normal datasets. This was, therefore, equivalent to simply regularizing the network by means of Eq. (5). That said, if the task of the machine learning model had been to predict future behavior of the building based on low quality input data, then the use of noise injection as data augmentation strategy should have led to considerably higher predictive performance.

Based on the previous considerations, this paper is not interested in using noise injection as a merely regularization approach. On the contrary, the objective is to bind the data augmentation strategy to the task, by performing denoising at the training, validation and evaluation stages. As such, it is aimed to learn a probability distribution $h = p(x)$ behind the original data, that is independent on the particular corrupted component x_i^*. This intermediate representation will be then exploited to recover the introduced data gaps, by means of a deterministic mapping $q(x|h)$.

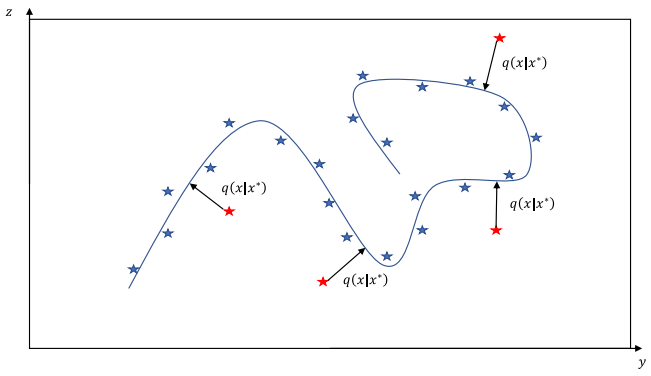


Fig. 1. An example of a one-dimensional manifold embedded in a two-dimensional space (y, z) with corrupted samples. Blue and red stars represent the uncorrupted and corrupted data samples, respectively. $q(x|x^*)$ is a stochastic operator that maps the corrupted samples to the manifold, during training.

3.3. Manifold learning

In the generic machine learning research, it is assumed that high dimensional data tend to concentrate in a low dimensional place called "manifold" [4]. Following this assumption, the main variations in the output of a learning model occur only on the data samples lying on that region [4]. Fig. 1 shows an example of a one-dimensional manifold embedded in a two-dimensional space. It can be noticed, that the corrupted data samples x^* (i.e. red stars on the picture) lie far from the curve. As such, the task of denoising can be re-interpreted from the point of view of manifold learning. During training, the model learns a manifold that is used to map a corrupted sample x^* to the original one x, by means of a stochastic operator $q(x|x^*)$. In particular, $q(x|x^*)$ is always pointing from x^* towards the closest point on the manifold. Hence, $q(x|x^*)$ is always orthogonal to the curve. Based on the previous considerations, the intermediate representation h, introduced in Section 3.2, can be regarded as a manifold [41].

According to the manifold assumption, in order to fulfill the task of denoising, a learning algorithm able to constrain the dimension of h is needed. Dimensionality reduction algorithms, such as autoencoders, are therefore selected for this study. As it is aimed to perform the task of denoising at the training, validation and evaluation stages, denoising autoencoders are chosen as modeling approach. For a detailed theoretical background about the used models, the reader is referred to Liguori et al. [34].

4. Datasets description

The datasets used in this paper collect 15-minutes time-step electric use data of three multi-family residential buildings located in the South-East area of Milan, for the year 2019. In particular, approximately 35,040 data points are available in each dataset. The registration refers to single flats and common areas, for a total of 151 data series. In particular, the three buildings have respectively 67 flats for a total gross surface of 4633 m^2, 57 flats for a total gross surface of 6260 m^2 and 27 flats for a total gross surface of 2049 m^2. The buildings were constructed between 1960 and 1980 and they were fully renovated (including systems and facades) around 2017.

The datasets are completely anonymous, without any type of information about tenants or the installed appliances. Specifically, the data is the accumulated electric absorption of all the electric uses in each flat; thus, the registration involves electric appliances and lighting, together with potential small space heating and/or cooling devices.

In the raw 15-minutes data, no errors or gaps are present. For each dataset, the descriptive statistics and buildings' ID can be observed in Table 3.

Table 3
Descriptive statistics. The standard deviation (Std) and the 75th and 25th percentiles are also given.

	Electricity consumption [Wh]		
	Dataset A	Dataset B	Dataset C
Max	44166.00	21664.00	18219.00
75th	15677.25	8863.00	9024.00
Median	10751.50	7126.00	6801.00
25th	7758.50	5699.00	3910.00
Min	0.00	0.00	0.00
Mean	11999.87	7180.05	6483.83
Std	4991.38	3232.39	3686.54

In the scope of this paper, the proposed data augmentation technique is coupled with an existing denoising autoencoder neural network. Therefore, in order to avoid additional architecture optimization, the described datasets should follow the same preprocessing procedure defined in the original paper. As explained by Liguori et al. [34], the existing models were implemented and optimized on 30 minutes-based daily time-series patterns, that were normalized using the Z-Score Normalization function. Therefore, the described datasets are also re-arranged into day-to-day matrices, downsampled to 30 min frequency and normalized using the same Z-Score Normalization function [34]. In order to obtain less biased training, validation and evaluation sets, the resulting matrices are then randomly shuffled by row one single time, using the related Numpy Python Library [45]. The effect of the random shuffling on the density distribution of different training sets can be observed in Section 5.2 (see Fig. 6). Finally, 10% of the total dataset, i.e. 36 days, is exported as a validation set for each dataset. For further information related to the adopted preprocessing procedure, the reader is referred to Liguori et al. [34]. The optimal training set rate is defined based on a minimization process described in the following section.

5. Methodology

In the scope of this paper, a data augmentation strategy based on masking noise injection is proposed to enhance the imputation accuracy of an existing denoising autoencoder. The case study consists in the data-efficient imputation of daily profiles of aggregate building energy time-series from three different residential buildings located in Milan, Italy (see Section 4). The aforementioned model was already implemented in a previous authors' work, for the reconstruction of missing IEQ data collected in a commercial building located in Aachen, Germany [34]. As observed by Lillstrang et al. [46], weather changes and occupant behavior have a well-known influence on the periodicity of building monitoring data. Generally, it is expected that indoor air temperature, CO_2 concentration and electricity consumption profiles will rise during the day and decrease during the night [46]. Although this consideration might not apply to households with atypical habits, aggregate electricity consumption data have much more regular daily profiles [47]. For that reason, denoising autoencoders implemented and optimized on daily profiles of IEQ data time-series were considered reasonable candidates for this study.

In the original work from Liguori et al. [34], convolutional, LSTM and feed-forward denoising autoencoders were implemented to reconstruct missing indoor air temperature (T), relative humidity (RH) and CO_2 concentration data. In particular, for all the analyzed data streams, each model was optimized differently in terms of hyperparameters. As such, nine different denoising autoencoders were implemented. The results proved that the convolutional variant outperformed all the other networks' architectures.

Based on the previous consideration, the existing denoising autoencoder was originally implemented for completely different tasks. Therefore, this model is selected in this paper to show how the proposed data augmentation process can improve the generalization capabilities of existing denoising autoencoders. A summary of the adopted methodology is presented in Fig. 2.

5.1. Model selection

As observed in Fig. 2, the first step of the adopted methodology consists of the selection of the optimal base imputation model from Liguori et al. [34]. Considering that, in the aforementioned paper, the convolutional variant outperformed all the other networks' architectures, it is chosen as the modeling approach of this study. The existing models were optimized differently on each data stream. Therefore, to avoid excessive performance drops, the denoising convolutional autoencoder that was optimized on the most similar data patterns is selected. This analysis is performed only on the training set of Dataset A, as it is aimed to evaluate the transferability of the results to the other case studies. In particular, the training set of Dataset A is initially set to 40% of the total data, i.e. 146 days.

In order to quantify the similarities between the target energy consumption and source indoor environmental quality daily profiles, Dynamic Time Warping (DTW) is applied [48]. The implementation is based on one of the related Python libraries [49]. The DTW is a widely used algorithm that can be used to compute the overall distance between two different temporal series [48]. As a rule of thumb, the higher the DTW distance, the lower the correlation between two different time-series. The similarity scores are computed between each row of the source and target matrices. The overall scores for the indoor air temperature, relative humidity and CO_2 concentration data are finally obtained with the mean operation and are given as follows: 5.96, 8.42 and 6.38, respectively. It can be therefore observed that the average distance between the normalized daily indoor air temperature and energy profiles is lower than all the other IEQ data streams. As such, the existing denoising convolutional autoencoder that was implemented for indoor air temperature daily profiles is selected. Additional information about the selected base imputation model can be found in the Appendix.

The normalized electricity consumption and IEQ daily average observations, drawn respectively from Dataset A and the Aachen dataset, are shown in Fig. 3. As expected, the indoor air temperature, CO_2 concentration and electricity consumption profiles rise during the day and decrease during the night. The indoor relative humidity daily average profile is significantly different from the other data streams. It is therefore expected to achieve similar performance when using the model optimized on indoor air temperature or CO_2 concentration data.

5.2. Data augmentation

The second step of the adopted methodology consists of the application of the proposed data augmentation strategy and the resulting minimization of the target training sets. Even in this case, the analysis is performed only on the Dataset A. In particular, it is aimed to minimize the initial 40% training set rate, by defining an optimal augmentation rate (Aug). Here, the augmentation rate is defined as the number of corrupted copies of the same training set that are stacked together to create an augmented training set. An example of the proposed data augmentation is shown in Fig. 4, where the same day of observation is corrupted two times in different positions (Aug = 2).

As observed in Fig. 4, the proposed data augmentation technique is significantly different from other well-known approaches, such as Gaussian noise and SMOTE [42]. The same applies to newly proposed approaches that increase the length and fine-grainedness of a small and low resolution dataset with synthetic interpolated data, such as fractal interpolation [50]. Here, the synthetic copies are created by just "masking" multiple times randomly selected components of the same training sample, according to Algorithms 1 and 2. As explained in Section 3, this is essential to boost the performance of a denoising autoencoder neural network used for missing data imputation.

Depending on the particular denoising task, the corrupted copies are derived by applying continuous or random missing masking noise with the same corruption rate, as described in Section 3.1. The optimized

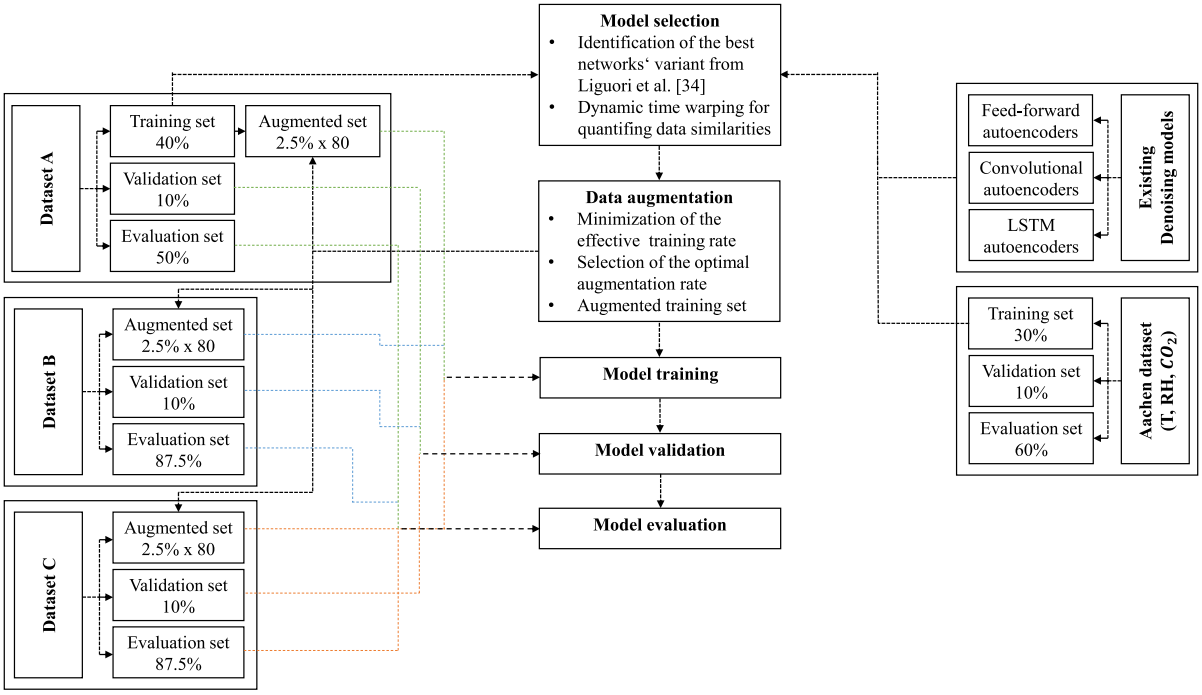


Fig. 2. Methodology overview. The percentages of the subsets of the Aachen dataset are the same as defined in Liguori et al. [34]. For consistency, the 10% validation set as defined in Liguori et al. [34] is also adopted for the other datasets. It is aimed to use Dataset A both for the initial analysis (in this Section) and for the final evaluation (Section 6). Hence, half of the dataset is exported for the evaluation phase. The other half (training + validation set) is used for the initial analysis. The values of the augmented sets are optimized in the following of this Section.

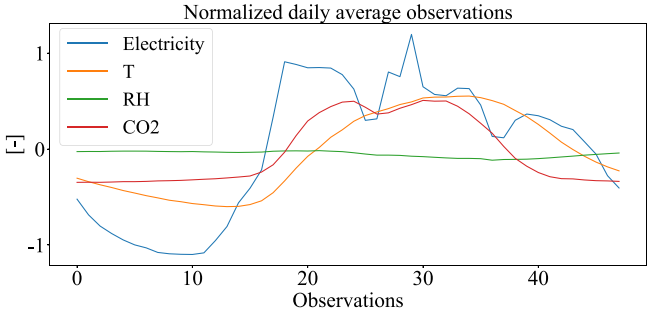


Fig. 3. Normalized daily average observations of electricity consumption and IEQ data, drawn respectively from Dataset A and the Aachen dataset.

training and augmentation rates are then exploited to augment the training sets of the remaining datasets.

In order to minimize the number of training data points from the target domain, the learning curves of the model are drawn for varying training and augmentation rates. In particular, for each missing scenario, the denoising autoencoder is trained from scratch with different training rates, ranging from 1.25% to 40%, i.e. from 4 to 146 days. Likewise, multiple augmentation rates are explored, ranging from 0 (no augmentation) to 80. The results are then shown on the validation set, by applying the root mean squared error (RMSE) to the real and reconstructed missing data. For the mathematical formulation of the RMSE, the reader is referred to Section 6.2. The learning curves for a corruption rate of 0.2 can be observed in Fig. 5.

It can be noticed that, for both continuous and random missing scenarios, there is a great improvement in the results. Augmenting the original training set can, indeed, lead to a 50% lower RMSE, occasionally, e.g. nine days-long training set with Aug equal to 80. In particular, the accuracy of the model improves by increasing both the training and augmentation rates. However, for a training set size larger than 36 days the augmented model does not perform significantly better. In order to investigate the previous consideration, the density distributions of different training sets for Dataset A are shown in Fig. 6.

It can be observed that starting from a training set size of 36 days, the density distributions do not change significantly anymore. As such, augmenting a 36 days-long training set has the same effect of augmenting a 146 days-long training set. The same consideration does not apply to the no augmentation case (Aug = 0), since more training data always lead to a better model's performance (see Fig. 5).

Based on the previous considerations, a training rate equal to or lower than 10% should be selected to minimize the amount of real data. Likewise, higher augmentation rates should be used to boost the performance of the model with fake data. In order to make a more comprehensive analysis, the execution time of the model is depicted in the form of "computational curves", for both continuous and random missing scenarios. Here, the execution time refers to the total time, in minutes, employed by the model for training and validation. The curves are shown in Fig. 7.

As expected, the behavior of the computational curves is generally opposite to the learning ones. As such, an augmentation rate of 80 leads to a much greater execution time compared to the no augmentation case. However, the difference between the two extreme cases decreases considerably for lower training rates. Therefore, the execution time of the model is neglected for this analysis.

In the end, the final training and augmentation rates are set to 2.5% (nine days) and 80, respectively. As previously observed in Fig. 5, augmenting 80 times a nine days-long training set can, indeed, lead to a 50% lower RMSE for both continuous and random missing scenarios. The latter values are therefore exploited to augment the training sets of the datasets, as shown in Fig. 2.

5.3. Model training, validation and evaluation

The last step of the adopted methodology consists in the evaluation of the existing denoising autoencoder, by using the previously defined augmented training sets. For that purpose, the model is trained and

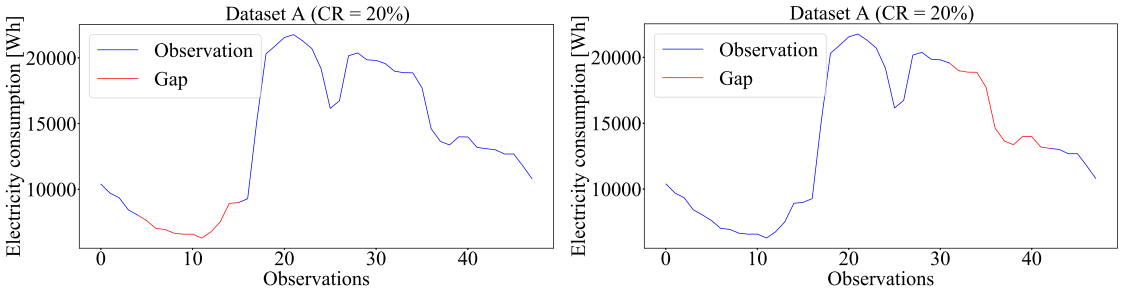


Fig. 4. An example of data augmentation with continuous missing masking noise injection. The same day of observation is corrupted two times with a corruption rate of 0.2.

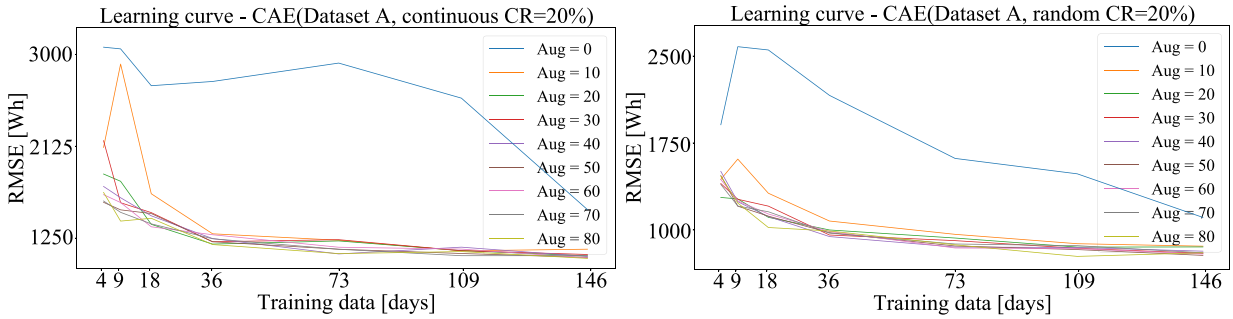


Fig. 5. Learning curves for the continuous (left) and random missing (right) scenarios with a corruption rate of 0.2, on Dataset A.

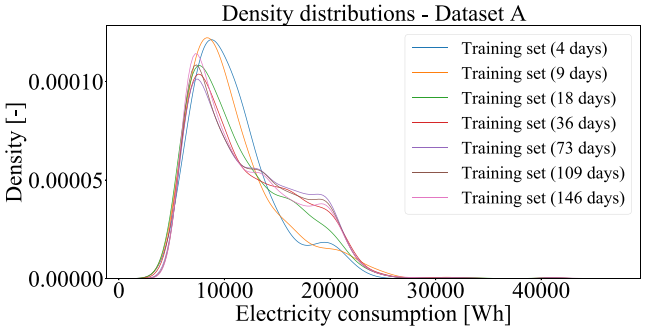


Fig. 6. Density distributions of different training sets for Dataset A.

evaluated separately for each dataset. Following the methodology proposed by Liguori et al. [34], the existing model is run 10 times on the augmented training set and validated on the validation set in order to address the stochastic initialization of weights. The best-performing model is eventually exported for further evaluation in the evaluation set. In particular, for Dataset A, the evaluation set consists of data that were not used during the previous analysis. As such, 50% of the dataset (183 days) is used for evaluation. For Dataset B and Dataset C, the evaluation set includes data that were not used for training and validation. Therefore, 87.5% of the dataset (319 days) is used for evaluation. The final results for the continuous and random missing scenarios are presented in the following Section.

6. Results

As previously explained, the proposed modeling approach is evaluated for both continuous and random missing scenarios. Before presenting the final results, the used benchmark models and performance evaluation metrics are introduced.

6.1. Benchmark models

The accuracy of the proposed modeling approach is compared to commonly applied statistical and machine learning-based imputation approaches. In particular, the Random Forest (RF), k-Nearest Neighbors (KNN) and linear (LI) and mean (MI) interpolation algorithms are defined as the benchmark models for this study. Further models that require excessive human expertise, computational resources or historical data are not selected due to the lack of comparability.

The first model is a machine learning-based imputation method based on the Random Forest algorithm, namely MissForest [51]. Consider a corrupted vector x^*, as a daily observation drawn from the evaluation set. The aforementioned sample has d missing components, where d depends on the corruption rate (see Section 3). The model exploits a training set consisting of t complete samples x, to train a RF model. At first, given a missing element $x^*_{miss_i}$, MissForest replaces the remaining missing components $x^*_{miss \neq miss_i}$ with their average values in the training set, as follows:

$$x^*_{miss_s \neq miss_i} = \frac{\sum_{j=1}^{t} x_{j,miss_s \neq miss_i}}{t}, \tag{6}$$

where $x^*_{miss_s \neq miss_i}$ is a missing element of x^* in position $miss_s$ other than $miss_i$ and $x_{j,miss_s \neq miss_i}$ is the existing element of the jth sample of the training set, in the same position. For each jth sample, the RF algorithm is trained to predict $x_{j,miss_i}$ based on the remaining elements of x_j. The trained RF is then used to impute $x^*_{miss_i}$ based on the remaining existing and averaged elements of x^*. This process is repeated for each missing component of x^*, in an iterative way. Therefore, at each iteration, the averaged and imputed components of x^* are replaced with newly imputed elements. Eventually, this process ends when a certain stopping criterion is met. The implementation of the MissForest algorithm is based on the related missingpy Python Library [52].

The second model is a machine learning-based imputation method based on the k-Nearest Neighbors algorithm, namely KNNimpute [53]. The model selects K complete samples x similar to x^*, from the training set. The reconstructed missing components are a weighted average of the existing values, as follows:

$$x^*_{miss_i} = \frac{\sum_{j=1}^{k} w_j x_{j,miss_i}}{k}, \tag{7}$$

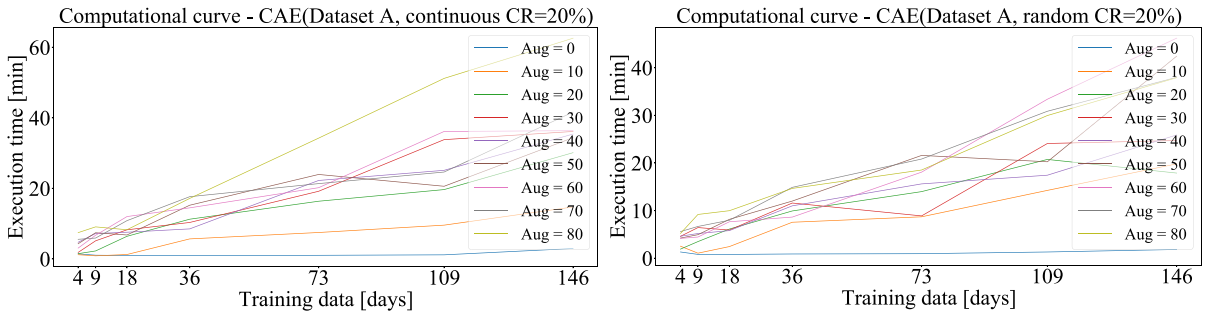


Fig. 7. Computational curves for the continuous (left) and random missing (right) scenarios with a corruption rate of 0.2, on Dataset A.

where w_j is a coefficient quantifying the similarity between the jth sample and x^*. The implementation of the aforementioned approach is based on the related Scikit-learn Python library [54]. In summary, KNNimpute combines different existing values from K similar samples into a final ensemble value, based on the average operation. However, it should be noted that the similarities metrics between the complete and corrupted samples are computed based on the Euclidean distance of the same variable, i.e. electricity consumption. This is one of the major differences[1] with respect to other methods that predict ensemble values based on the average operation [55]. For example, Lotfi et al. [55] defines an ensemble algorithm for solar power forecasting based on the Kernel Density Estimation (KDE) similarity metric. As for the KNNimpute algorithm, the model selects K samples x similar to x^*, where x^* is the target sample. However, here, x consists of several weather and calendar variables related to the specific solar power measurement. Furthermore, several bandwidth coefficients have to be manually tuned for computing the KDE-based similarity metrics. As the aforementioned bandwidth values quantify the level of correlation between each variable of x and the related solar power data, this last step requires specific human expertise. For that reason, KNNimpute is preferred to the described KDE-based ensemble method. In particular, in order to allow a fair comparison with the denoising autoencoder, both the proposed and benchmark models are trained on the same training set.

The third model is the linear interpolation algorithm between the last two known values of a data gap. Given two existing values x^*_{left} and x^*_{right}, the missing intermediate component $x^*_{miss_i}$ is imputed by means of the following equation:

$$x^*_{miss_i} = \frac{obs_{miss_i} - obs_{left}}{obs_{right} - obs_{left}}(x^*_{right} - x^*_{left}) + x^*_{left} \ , \tag{8}$$

where obs is the observation time.

The last model is the mean interpolation algorithm between the last two known values of a data gap. The missing intermediate component is obtained as follows:

$$x^*_{miss_i} = \frac{x^*_{left} + x^*_{right}}{2} \ . \tag{9}$$

6.2. Performance evaluation metrics

In order to assess the imputation accuracy of the previous methods, the RMSE, normalized RMSE (NRMSE) and mean absolute error (MAE) are selected as the performance metrics of this study. Since the denoising autoencoder is trained to reconstruct corrupted days of observation, the aforementioned metrics are applied to the corrupted values of each sample (row of a matrix). Eventually, the overall error is computed by means of the mean operation. The RMSE equation for the jth sample is given as follows:

$$RMSE_j = \sqrt{\frac{\sum_i^n (x_{j,i} - x_{j,i}^{imp})^2}{n}} \ , \tag{10}$$

where x and x^{imp} are respectively the real and imputed samples. Additionally, the number of corrupted components for each sample is defined by n and the position of the corrupted component is defined by i. The overall RMSE is given as follows:

$$RMSE = \frac{\sum_{j=1}^m RMSE_j}{m} \ , \tag{11}$$

where m defines the number of evaluated samples.

As explained by Liguori et al. [56], the RMSE alone does not provide sufficient information about the average model's accuracy. Being a quadratic metric, it tends indeed to penalize larger errors. For that reason, the MAE is selected as additional absolute performance metric. The MAE equation for the jth sample is given as follows:

$$MAE_j = \frac{\sum_i^n |x_{j,i} - x_{j,i}^{imp}|}{n} \ . \tag{12}$$

The overall MAE is then computed as before:

$$MAE = \frac{\sum_{j=1}^m MAE_j}{m} \ . \tag{13}$$

Finally, a comparison between different datasets is made by means of the NRMSE, defined as in [34]:

$$NRMSE = \frac{RMSE}{IQR} \ , \tag{14}$$

where IQR is the interquartile range, defined as the difference between the 75th and 25th percentiles of the dataset.

6.3. Continuous missing scenario

Table 4 summarizes the aforementioned performance evaluation metrics at different corruption rates and buildings, for continuous missing. In particular, the corruption rates are varied from 0.2 to 0.8. As such, from 5 to 19 h of data are missing in each sample. The accuracy of the proposed modeling approach is compared to the denoising autoencoder without augmentation and to the benchmark models explained in Section 6.1. Additionally, the average results over the analyzed buildings and corruption rates are shown in Table 5.

In terms of NRMSE, the denoising convolutional autoencoder with augmented training set (CAE + Aug) can impute the continuous missing data with the highest accuracy on Dataset C. The lowest accuracy is observed on Dataset B. In particular, the average NRMSEs on Dataset A, Dataset B and Dataset C are 0.32, 0.36 and 0.25, respectively. Nonetheless, the absolute performance metrics indicate a different behavior of the model. The average RMSEs are 2586.47 Wh, 1141.26 Wh and 1306.26 Wh, respectively. Likewise, the average MAEs are 2071.48 Wh, 941.08 Wh and 1074.50 Wh, respectively. The difference between

[1] An other difference is the use of a confidence interval around the expected output.

Table 4
Performance evaluation metrics at different corruption rates for continuous missing scenario.

Continuous missing	CR [–]	Dataset A					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	0.20	3578.60	1642.00	2328.96	1863.13	2229.73	2701.48
	0.40	4225.85	2354.47	2797.05	2544.63	3165.35	3764.50
	0.60	4162.90	3002.15	3246.09	3444.63	4362.29	4886.02
	0.80	3914.16	3347.27	3373.71	3671.87	4572.89	5153.85
MAE [Wh]	0.20	3175.55	1346.38	2015.54	1580.17	1860.13	2349.58
	0.40	3763.32	1867.43	2346.25	2082.57	2573.37	3291.07
	0.60	3642.91	2422.38	2691.25	2777.92	3570.15	4302.55
	0.80	3136.78	2649.72	2707.00	2840.71	3653.47	4483.83
NRMSE [–]	0.20	0.45	0.20	0.29	0.23	0.28	0.34
	0.40	0.53	0.29	0.35	0.32	0.40	0.47
	0.60	0.52	0.38	0.41	0.43	0.55	0.61
	0.80	0.49	0.42	0.42	0.46	0.57	0.65
	CR [–]	Dataset B					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	0.20	1511.92	982.21	1788.07	1535.97	1572.89	1730.40
	0.40	1703.99	1145.08	1847.08	1613.00	1822.76	2007.18
	0.60	2010.03	1159.71	1881.28	1781.77	1877.96	2170.15
	0.80	1766.48	1278.03	1871.12	2064.56	2163.50	2656.65
MAE [Wh]	0.20	1318.36	831.58	1625.96	1380.67	1442.14	1595.95
	0.40	1411.81	955.66	1641.35	1416.48	1634.19	1838.61
	0.60	1748.81	946.81	1658.40	1558.73	1657.02	1921.29
	0.80	1436.85	1030.27	1626.04	1808.32	1937.94	2347.35
NRMSE [–]	0.20	0.47	0.31	0.56	0.48	0.49	0.54
	0.40	0.54	0.36	0.58	0.51	0.57	0.63
	0.60	0.63	0.36	0.59	0.56	0.59	0.68
	0.80	0.56	0.40	0.59	0.65	0.68	0.84
	CR [–]	Dataset C					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	0.20	2543.11	1202.95	2186.66	1437.44	1468.61	1634.66
	0.40	1949.44	1247.29	2254.45	1817.20	1769.77	1852.12
	0.60	2132.00	1350.88	2308.01	2091.13	1760.78	1848.68
	0.80	2276.11	1423.93	2290.09	2328.94	1940.07	2303.65
MAE [Wh]	0.20	2330.22	1015.47	2023.19	1282.72	1254.42	1446.10
	0.40	1630.59	1022.29	2061.04	1620.94	1472.64	1585.99
	0.60	1770.60	1099.92	2096.93	1874.44	1424.09	1547.45
	0.80	1906.07	1160.33	2050.93	2077.12	1603.17	1954.29
NRMSE [–]	0.20	0.49	0.23	0.42	0.28	0.28	0.32
	0.40	0.38	0.24	0.44	0.35	0.34	0.36
	0.60	0.41	0.26	0.45	0.41	0.34	0.36
	0.80	0.44	0.28	0.44	0.45	0.38	0.45

Table 5
Average performance evaluation metrics for continuous missing scenario.

Continuous missing	Building ID	Average					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	Dataset A	3970.38	2586.47	2936.45	2881.06	3582.57	4126.46
	Dataset B	1748.11	1141.26	1846.89	1748.82	1859.28	2141.10
	Dataset C	2225.17	1306.26	2259.80	1918.67	1734.81	1909.78
	Tot. Average	2647.88	1678.00	2347.71	2182.85	2392.22	2725.78
MAE [Wh]	Dataset A	3429.64	2071.48	2440.01	2320.34	2914.28	3606.76
	Dataset B	1478.96	941.08	1637.94	1541.05	1667.82	1925.80
	Dataset C	1909.37	1074.50	2058.02	1713.80	1438.58	1633.46
	Tot. Average	2272.66	1362.35	2045.32	1858.40	2006.89	2388.67
NRMSE [–]	Dataset A	0.50	0.32	0.37	0.36	0.45	0.52
	Dataset B	0.55	0.36	0.58	0.55	0.58	0.67
	Dataset C	0.43	0.25	0.44	0.37	0.34	0.37
	Tot. Average	0.49	0.31	0.46	0.42	0.46	0.52

the normalized and absolute metrics suggests high data variability in Dataset A. Likewise, low data variability is expected in Dataset B. The previous consideration can be observed in the descriptive statistics of the analyzed datasets, shown in Table 3.

The average performance of the base imputation model increases considerably by training the model on the augmented data. In particular, the average RMSE and MAE are 36.66% and 40.05% lower than the no augmentation case (CAE). This difference is even more remarkable for lower corruption rates, where the average RMSE and MAE decrease by 47.28% and 50.31%, respectively. The augmentation

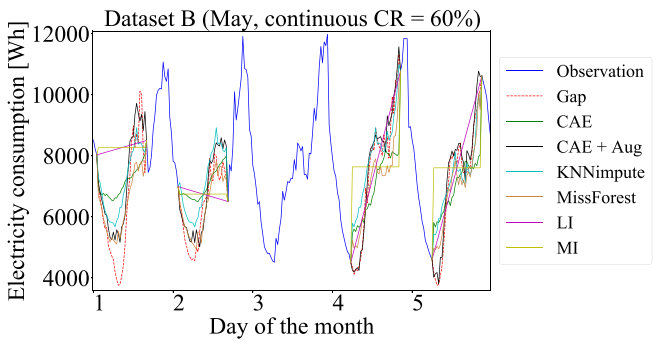


Fig. 8. Exemplary continuous missing imputation process for 5 days-long electricity consumption data selected from Dataset B in May, with a corruption rate of 0.6.

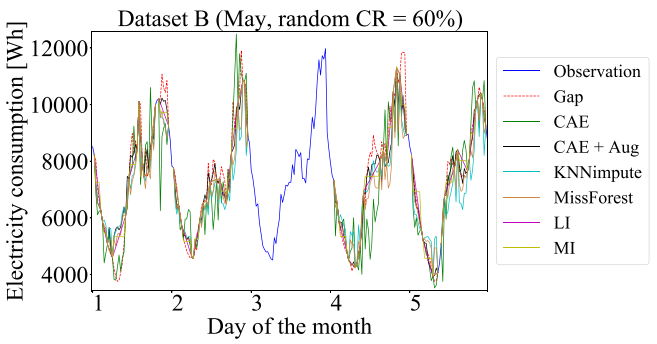


Fig. 9. Exemplary random missing imputation process for 5 days-long electricity consumption data selected from the Dataset B in May, with a corruption rate of 0.6.

process becomes less effective for higher corruption rates, where the average RMSE and MAE decrease by 26.52% and 27.64%, respectively. This different performance enhancement can be explained considering that, for each sample, fewer combinations of missing components are possible by increasing the corruption rate.

The proposed modeling approach outperforms all analyzed benchmark models. In particular, the average RMSE and MAE are respectively almost 23.13% and 26.69% lower than the best performing imputation method, namely MissForest. This difference tends to decrease for lower corruption rates and on Dataset A. Fig. 8 shows an exemplary continuous missing imputation process for 5 days-long electricity consumption data, selected from Dataset B in May. The presented data are corrupted with a corruption rate of 0.6, which means that 14.50 h of data are missing in each day of observation. Non-corrupted days are not part of the evaluation set. It can be observed that the simplified statistical methods are usually not able to follow the real trend of the data gap. Additionally, the CAE + Aug method tends to generally follow the peaks. In conclusion, the CAE + Aug method is the only imputation approach able to fully capture the real daily profiles of energy consumption data.

6.4. Random missing scenario

Table 6 summarizes the performance evaluation metrics at different corruption rates and buildings, for random missing. As before, the corruption rates varied from 0.2 to 0.8. The accuracy of the proposed modeling approach is compared to the denoising autoencoder without augmentation and to the benchmark models explained in Section 6.1. The average results over the analyzed buildings and corruption rates are shown in Table 7.

In terms of NRMSE, the denoising convolutional autoencoder with augmented training set can impute the random missing data with higher accuracy than in the continuous missing scenario. In particular, the average NRMSEs on Dataset A, Dataset B and Dataset C are 0.20, 0.26 and 0.21, respectively. So, similarly to the continuous missing case, the lowest accuracy is observed on the Dataset B. The average RMSEs are 1589.16 Wh, 839.98 Wh and 1072.77 Wh, respectively. The average MAEs are 1177.52 Wh, 666.42 Wh and 859.09 Wh, respectively. Therefore, the absolute performance metrics confirm the consideration made in the previous section, i.e. lower data variability is expected in Dataset B.

The average performance of the base imputation model increases even more than in the continuous missing scenario, by training the model on the augmented data. This can be explained considering that, for each sample, more combinations of missing components can be explored by applying random rather than continuous missing masking noise. In particular, the average RMSE and MAE are 47.90% and 51.19% lower than the no augmentation case, respectively.

The proposed modeling approach outperforms all the analyzed benchmark models. However, the imputation performance of the simplified statistical models is significantly better than in the continuous missing scenario. This performance enhancement is expected, since the random missing imputation is a relatively simpler task than continuous missing imputation, as the non-corrupted entries are usually closer to the corrupted one (see Section 3.1). In particular, the average RMSE and MAE of the augmented model are respectively 12.09% and 11.99% lower than the best performing imputation method, i.e. LI. The difference in accuracy between the last methods tends to increase for higher corruption rates and on Dataset B. On the contrary, it decreases significantly on Dataset A and Dataset C. This last consideration suggests that datasets with lower data variability might be more difficult to impute by using simplified statistical approaches. Fig. 9 shows an exemplary random missing imputation process for five days-long electric consumption data, selected from the Dataset B in May. The presented data are corrupted with a corruption rate of 0.6. It can be observed that the KNNimpute and MissForest methods generally fail to reconstruct the missing values.

7. Discussion

7.1. Real vs reconstructed datasets

In order to quantify the influence of the proposed augmented imputation method on the descriptive statistics of the evaluation sets, the density distributions are shown in Fig. 10. This is important to determine if the reconstructed datasets might be biased towards some specific values. In particular, the analysis is performed for each building and missing scenario, by varying the corruption rate.

In general, the density distributions of the reconstructed evaluation sets are strongly resemble to the real ones. However, high corruption rates might introduce bias in the results. This behavior could be explained by considering that autoencoder neural networks are often affected by saturation issues. As such, if network saturation occurs, data gaps are always imputed with the same value. Even if network saturation is never achieved in this study, the results from Liguori et al. [34] demonstrates that the used model can generally be closer to a saturation condition when the corruption rate is increased.

7.2. Impact of different periods

The impact of different periods of observation on the performance of the denoising convolutional autoencoder with augmented training set is shown in Fig. 11. In particular, the average RMSEs for the continuous and random missing scenarios (see Tables 5 and 7) are distributed over different months and days of the week.

The distribution of the average RMSE over the different months of the year shows that colder periods have a negative effect on the

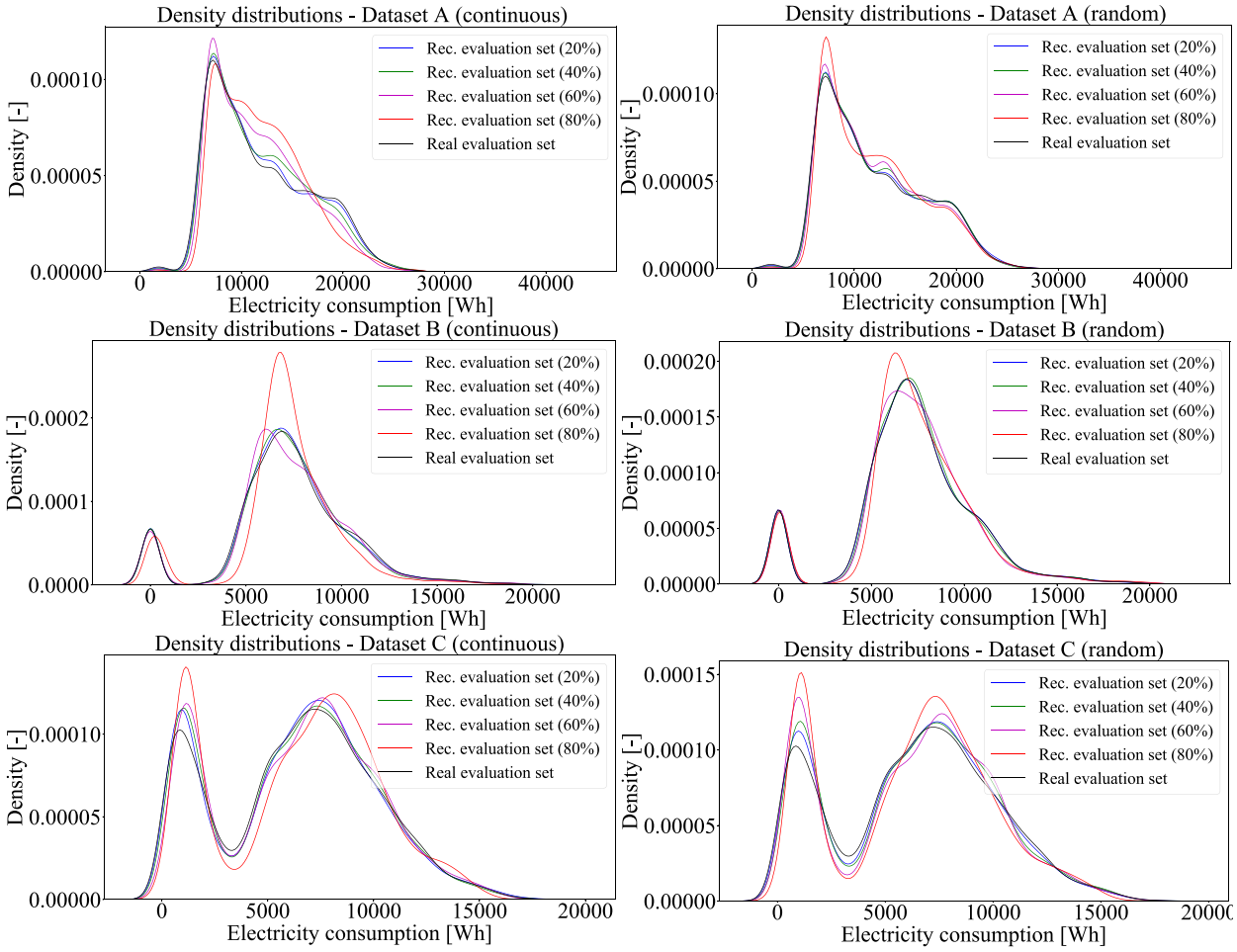


Fig. 10. Density distributions for the real and imputed evaluation sets, at different corruption rates. Dataset A, Dataset B and Dataset C can be observed respectively on the first, second and third row. Continuous and random missing scenarios on the first and second column.

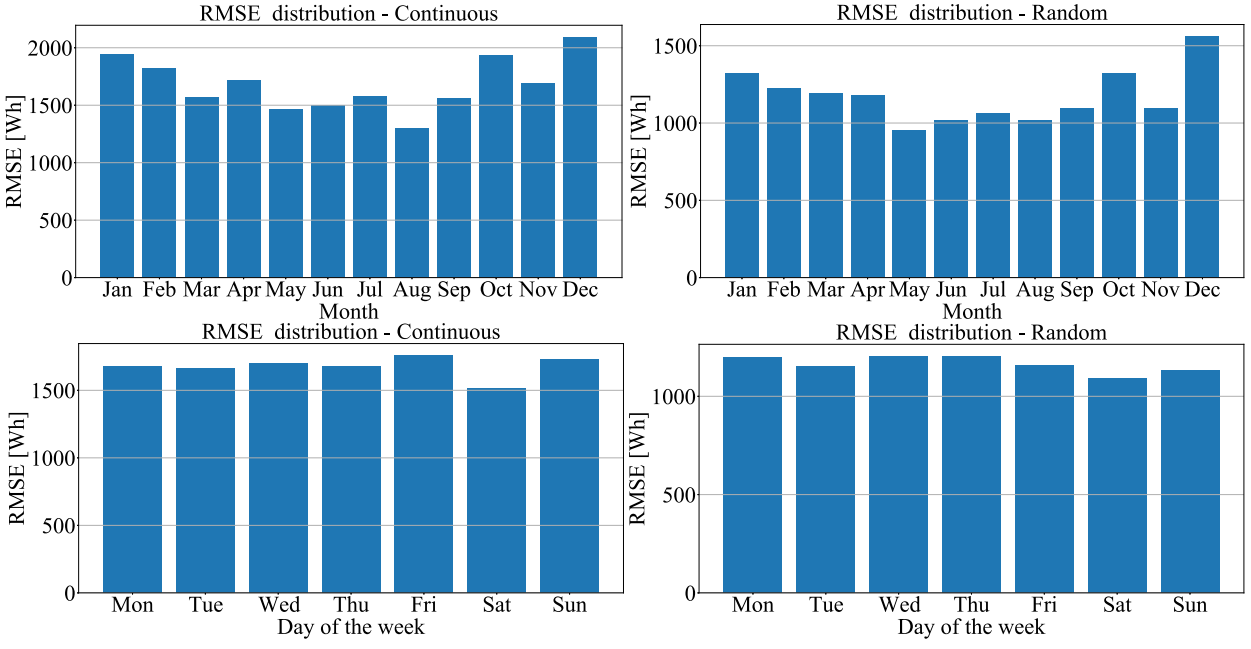


Fig. 11. Average RMSEs for continuous (left) and random (right) missing scenarios, distributed over different months (top) and days of the week (bottom).

Table 6
Performance evaluation metrics at different corruption rates for random missing scenario.

Random missing	CR [–]	Dataset A					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	0.20	2688.70	1355.89	2160.10	1625.56	1104.91	1189.39
	0.40	2659.44	1376.18	2210.95	1816.12	1323.49	1456.32
	0.60	3362.83	1623.74	2244.20	2135.08	1643.07	1821.39
	0.80	3807.99	2000.82	2325.06	2827.34	2366.54	2746.60
MAE [Wh]	0.20	2233.64	1035.03	1729.83	1230.96	763.73	814.00
	0.40	2123.95	1010.73	1741.58	1356.95	873.35	974.11
	0.60	2741.86	1190.29	1757.84	1573.89	1108.52	1265.77
	0.80	3008.90	1474.01	1815.64	2112.63	1714.40	2067.84
NRMSE [–]	0.20	0.34	0.17	0.27	0.20	0.14	0.15
	0.40	0.33	0.17	0.28	0.23	0.16	0.18
	0.60	0.42	0.20	0.28	0.27	0.20	0.23
	0.80	0.48	0.25	0.29	0.35	0.30	0.34
	CR [–]	Dataset B					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	0.20	1160.39	745.38	1799.30	1488.52	1143.15	1155.43
	0.40	1407.64	741.87	1813.16	1546.56	1196.93	1241.87
	0.60	1697.95	892.83	1813.66	1660.23	1322.63	1406.86
	0.80	1690.08	979.85	1813.09	2014.51	1636.17	1770.35
MAE [Wh]	0.20	956.20	596.40	1566.49	1289.37	1031.86	1042.26
	0.40	1140.10	587.64	1568.74	1327.87	1067.10	1102.31
	0.60	1351.45	701.23	1563.09	1419.74	1161.58	1231.73
	0.80	1374.57	780.41	1561.01	1741.47	1413.12	1542.81
NRMSE [–]	0.20	0.36	0.23	0.57	0.47	0.36	0.36
	0.40	0.44	0.23	0.57	0.49	0.38	0.39
	0.60	0.53	0.28	0.57	0.52	0.42	0.44
	0.80	0.53	0.31	0.57	0.63	0.51	0.56
	CR [–]	Dataset C					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	0.20	1623.38	990.54	2174.87	1446.19	791.83	809.04
	0.40	1768.34	1019.97	2203.83	1787.85	878.35	926.62
	0.60	2447.40	1100.23	2210.72	2027.17	1041.93	1152.42
	0.80	2575.12	1180.34	2237.79	2300.05	1485.68	1602.91
MAE [Wh]	0.20	1323.77	802.28	1970.25	1241.69	592.80	609.41
	0.40	1476.74	814.81	1985.27	1563.91	649.80	696.49
	0.60	2160.59	878.30	1986.01	1796.00	776.58	887.20
	0.80	2261.45	940.95	2004.36	2053.12	1132.53	1281.66
NRMSE [–]	0.20	0.31	0.19	0.42	0.28	0.15	0.16
	0.40	0.34	0.20	0.43	0.35	0.17	0.18
	0.60	0.48	0.21	0.43	0.39	0.20	0.22
	0.80	0.50	0.23	0.43	0.45	0.29	0.31

Table 7
Average performance evaluation metrics for random missing scenario.

Random missing	Building ID	Average					
		CAE	CAE + Aug	KNNimpute	MissForest	LI	MI
RMSE [Wh]	Dataset A	3129.74	1589.16	2235.08	2101.02	1609.50	1803.43
	Dataset B	1489.02	839.98	1809.80	1677.45	1324.72	1393.63
	Dataset C	2103.56	1072.77	2206.80	1890.31	1049.45	1122.75
	Tot. Average	2240.77	1167.30	2083.89	1889.60	1327.89	1439.93
MAE [Wh]	Dataset A	2527.09	1177.52	1761.22	1568.60	1115.00	1280.43
	Dataset B	1205.58	666.42	1564.83	1444.61	1168.42	1229.78
	Dataset C	1805.64	859.09	1986.47	1663.68	787.93	868.69
	Tot. Average	1846.10	901.01	1770.84	1558.96	1023.78	1126.30
NRMSE [–]	Dataset A	0.39	0.20	0.28	0.26	0.20	0.23
	Dataset B	0.47	0.26	0.57	0.52	0.42	0.44
	Dataset C	0.41	0.21	0.43	0.36	0.20	0.22
	Tot. Average	0.42	0.22	0.43	0.38	0.27	0.29

imputation accuracy of the proposed method. In particular, for both continuous and random missing scenarios, the month of the year with the highest RMSE is December. On the contrary, the months of the year with the lowest RMSE are August and May, respectively for continuous and random missing. This behavior might be explained by considering that the average occupancy profiles of residential buildings might decrease during the summer period, e.g. summer holidays and outdoor activities, reducing the electric consumption data variability.

In order to perform a more detailed analysis, the daily average profiles over each month of the year are depicted in Fig. 12. The most atypical pattern is noticed in December, which is characterized by

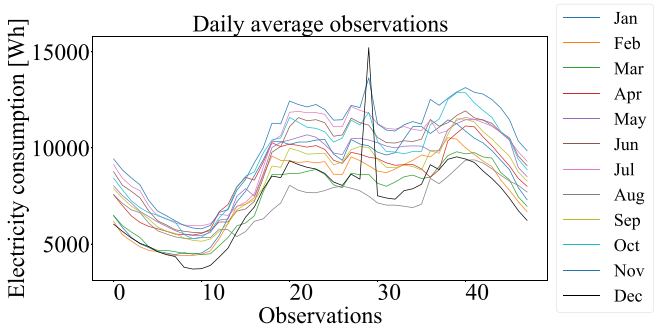


Fig. 12. Daily average observations distributed over different months.

significant peaks of electricity consumption. The same month is characterized by much lower average observations compared to January or November, which suggests the presence of anomalies in the data. Additionally, it is observed that different months have different scales. As such, it is expected that absolute metrics such as the RMSE would generally give smaller errors for months of the year with lower scales, such as August.

The distribution of the average RMSE over different days shows that the imputation accuracy of the proposed method is slightly affected during the week. In particular, for both continuous and random missing scenarios, the day of the week with the lowest RMSE is Saturday. On the contrary, the days of the week with the highest RMSE are Friday and Thursday, respectively for continuous and random missing.

The previous analysis might be useful for energy modelers, in order to decide when to apply the proposed augmented imputation method. In summary, it is observed that periods of the year with higher data variability and peaks might be, in general, more challenging to reconstruct. However, future work should analyze the impact of different periods on buildings with different occupancy patterns, such as commercial or office buildings.

7.3. Impact of different levels of data aggregation and peak electricity consumption

The results from Section 6 indicate that augmenting a small training set with masking noise is a suitable approach for improving the imputation accuracy of an existing convolutional denoising autoencoder. Additionally, experiments with electricity consumption data aggregated at the building level demonstrate that the proposed method outperforms commonly applied statistical and machine learning-based imputation approaches.

Typically, aggregate electricity consumption data have much more regular and predictable daily profiles than single users' [47]. In this regard, it is important to evaluate the proposed method in the case of disaggregate data. As described in Section 4, each building has 67 flats, each of which provides data consisting of accumulated electric absorption of all the electric users. In this Section, a preliminary analysis is performed by considering only one exemplary flat from each building. Following the methodology described in Section 5, nine complete days of observations are exported from each dataset and augmented 80 times for training. The results for continuous and random missing scenarios can be observed in Table 8.

The average RMSEs and MAEs of the convolutional denoising autoencoder with augmented training set are much lower at the disaggregate flat level. However, the absolute metrics cannot be used to compare datasets with different scales. For that reason, only the results obtained from normalized metrics are analyzed. For the continuous missing scenario at the flat level, the average NRMSEs on Dataset A, Dataset B and Dataset C are 0.67, 0.68 and 1.12, respectively. Therefore the average NRMSE is around 182% higher than the building-level case. For the random missing scenario at the flat-level, the average NRMSEs on Dataset A, Dataset B and Dataset C are 0.56, 0.64 and 0.95, respectively. Therefore the average NRMSE is around 226% higher than the building-level case. Based on the previous considerations, it is observed that the denoising convolutional autoencoder with augmented training set performs poorly at the disaggregate flat level. This can be explained by considering that the existing base imputation model was optimized on daily profiles of indoor air temperature data. As presented in Section 5.1, aggregate electricity consumption data have much more similar profiles than disaggregate ones. Therefore, the base imputation model should encounter additional architecture optimization, in order to achieve satisfactory performance.

Fig. 13 shows exemplary continuous and random missing imputation processes for 5 days-long electricity consumption data. The case study is a single flat from Dataset B in May, with a corruption rate of 0.6. Compared to the building level case depicted in Figs. 8 and 9, the daily profiles appear much more irregular and are characterized by higher peaks. This last observation suggests that the reconstruction error might be higher when imputing peaks of electricity consumption.

In order to strengthen the previous claim, the proposed method is further evaluated for reconstructing peaks electricity consumption. For simplicity, the daily profiles of the evaluation set are corrupted during the five hours of maximum average electricity consumption. This corresponds to a continuous missing scenario with a corruption rate of 0.2 on the highest peak of the day. The results at the aggregate and disaggregate levels are presented in Table 9.

At the building level, the average NRMSE at the peak is 39% higher than the usual continuous scenario. On the other hand, at the flat level, the average NRMSE at the peak is 73% higher than the usual continuous scenario. Therefore, the results suggest that the disaggregate electricity consumption profiles are characterized by much higher peaks that further degrade the performance of the model. This further degradation of performance on the peak electricity consumption might be explained by considering that the proposed data augmentation technique creates synthetic copies by just "masking" multiple times randomly selected components of the same training sample. Therefore, a few combinations of missing components at the peak can be explored during training.

Future work should analyze more in detail the impact of different levels of data aggregation and peak electricity consumption on the proposed method. In this regard, the base imputation model should encounter additional architecture optimization, to achieve satisfactory performance.

7.4. Computational requirements

The execution time of the augmented imputation method was already analyzed in Section 5.2. It was proven that, for training and augmentation rates equal to respectively 2.5% and 80, the time required for training and validation was approximately 10 min. In order to facilitate the integration of the method into real-time building control applications, an estimate of the computational requirements of the pre-trained model is required. As described by Dong et al. [27], the computational requirements should, therefore, refer to the model evaluation, rather than to training and validation. For that purpose, as suggested in the same paper, the inference runtime and inference memory requirements are evaluated. To document the scalability of the proposed method, these are evaluated for different numbers of samples. In particular, the maximum number of samples is defined by the length of the smallest evaluation set, i.e. 183 days for the Dataset A. The average results over the analyzed buildings can be observed in Fig. 14.

The relation between the inference time and the number of evaluated samples is almost linear. Additionally, there is no clear difference between varied corruption rates or missing scenarios. The average inference time of one sample is 0.0025 s. On the other hand, the average inference time of 183 samples is 0.077 s. Finally, there is an additional increase in inference time after every 32 samples. This

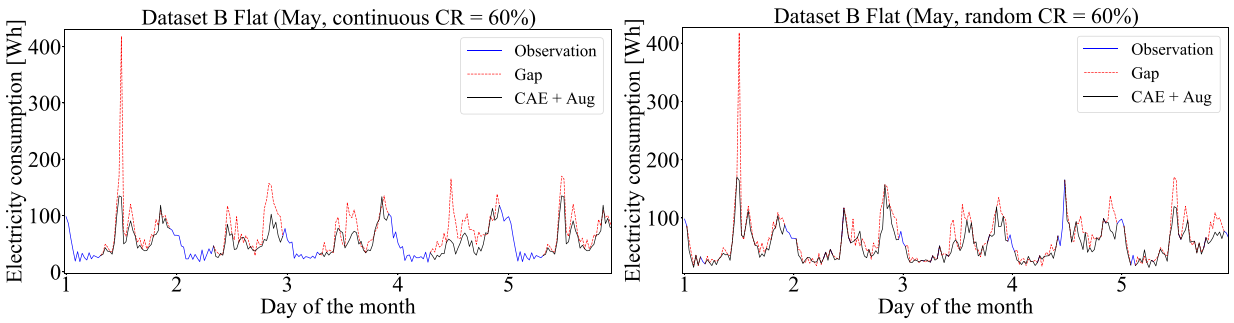


Fig. 13. Exemplary continuous (left) and random (right) missing imputation process for 5 days-long electricity consumption data. The case study is a single flat from Dataset B in May, with a corruption rate of 0.6.

Table 8
Performance evaluation metrics at different corruption rates and levels of data aggregation for continuous (top) and random missing (bottom) scenarios.

Continuous missing	CR [–]	Dataset A		Dataset B		Dataset C	
		Building	Flat	Building	Flat	Building	Flat
RMSE [Wh]	0.20	1642.00	24.95	982.21	24.86	1202.95	120.30
	0.40	2354.47	25.58	1145.08	27.18	1247.29	141.64
	0.60	3002.15	28.28	1159.71	27.90	1350.88	147.39
	0.80	3347.27	29.43	1278.03	28.97	1423.93	149.31
	Average	2586.47	27.06	1141.26	27.23	1306.26	139.66
MAE [Wh]	0.20	1346.38	21.15	831.58	19.24	1015.47	93.52
	0.40	1867.43	20.76	955.66	19.28	1022.29	107.64
	0.60	2422.38	22.88	946.81	19.05	1099.92	106.99
	0.80	2649.72	23.66	1030.27	19.55	1160.33	94.29
	Average	2071.48	22.11	941.08	19.28	1074.50	100.61
NRMSE [–]	0.20	0.20	0.62	0.31	0.62	0.23	0.97
	0.40	0.29	0.64	0.36	0.68	0.24	1.14
	0.60	0.38	0.70	0.36	0.69	0.26	1.18
	0.80	0.42	0.73	0.40	0.72	0.28	1.20
	Average	0.32	0.67	0.36	0.68	0.25	1.12
Random missing	CR [–]	Dataset A		Dataset B		Dataset C	
		Building	Flat	Building	Flat	Building	Flat
RMSE [Wh]	0.20	1355.89	19.99	745.38	24.16	990.54	108.12
	0.40	1376.18	20.60	741.87	24.94	1019.97	112.11
	0.60	1623.74	23.15	892.83	25.99	1100.23	120.29
	0.80	2000.82	26.09	979.85	27.15	1180.34	131.04
	Average	1589.16	22.46	839.98	25.56	1072.77	117.89
MAE [Wh]	0.20	1035.03	16.06	596.40	17.57	802.28	76.65
	0.40	1010.73	16.25	587.64	17.11	814.81	77.52
	0.60	1190.29	17.79	701.23	17.79	878.30	83.35
	0.80	1474.01	20.12	780.41	18.35	940.95	84.60
	Average	1177.52	17.56	666.42	17.71	859.09	80.53
NRMSE [–]	0.20	0.17	0.50	0.23	0.60	0.19	0.87
	0.40	0.17	0.51	0.23	0.62	0.20	0.90
	0.60	0.20	0.58	0.28	0.65	0.21	0.97
	0.80	0.25	0.65	0.31	0.67	0.23	1.05
	Average	0.20	0.56	0.26	0.64	0.21	0.95

Table 9
Performance evaluation metrics at varied levels of data aggregation, for peak and continuous missing scenarios. The analyzed corruption rate is 0.2.

		Dataset A		Dataset B		Dataset C	
		Building	Flat	Building	Flat	Building	Flat
RMSE [Wh]	Peak missing	2327.88	35.53	1504.82	46.41	1449.10	227.39
	Continuous missing	1642.00	24.95	982.21	24.86	1202.95	120.30
MAE [Wh]	Peak missing	1847.91	29.56	1300.06	35.33	1231.31	177.97
	Continuous missing	1346.38	21.15	831.58	19.24	1015.47	93.52
NRMSE [–]	Peak missing	0.29	0.89	0.47	1.16	0.28	1.83
	Continuous missing	0.20	0.62	0.31	0.62	0.23	0.97

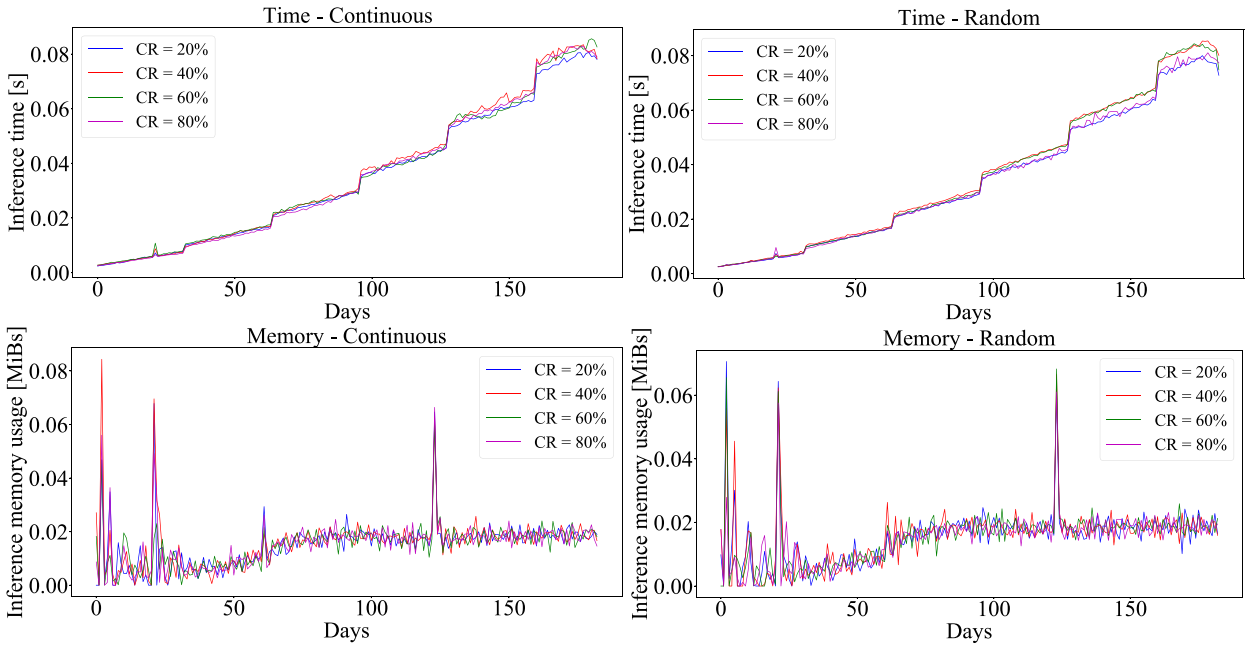


Fig. 14. Average inference time (top) and memory usage (bottom) of different numbers of evaluation samples at different corruption rates, for continuous (left) and random (right) missing scenarios.

repeated loss of computational efficiency might be reduced by adapting the batch size of the model to the number of evaluated samples.

The inference memory usage generally increases with the number of evaluated samples. However, it achieves a constant average value of 0.018 MiBs after the 75th day of evaluation. Peaks of memory usage up to 0.086 MiBs can sporadically occur during inference. Finally, there is no clear difference between varied corruption rates or missing scenarios.

Simulations were performed with computing resources granted by RWTH Aachen University, using a Linux CentOS 7.9 operating system. In particular, the used computing nodes were equipped with two NVIDIA Volta V100 GPUs and connected to an Intel Omni-Path 100 Gigabits network. For additional information regarding the adopted computational environment, the reader is referred to the description of the Tier 2 System CLAIX-2018 on the RWTH website [57].

8. Conclusion

This paper presented and validated a simple, yet effective, methodology to impute missing building energy data time-series with existing denoising autoencoder neural networks. The presented solution can be easily adopted by building practitioners, as it does not require excessive human expertise, computational resources or historical data. Therefore, it is widely applicable to different settings.

The proposed methodology relies on a particular data augmentation technique which consists in stacking together multiple synthetic copies of a relatively small training dataset with pseudo-random noise. The base imputation model is a denoising convolutional autoencoder implemented in a previous study, for IEQ data reconstruction. However, the presented method could be coupled with any existing denoising autoencoder implemented for generic time-series that share similar daily periodicity and variability. In particular, in order to simulate different missing scenarios, continuous and random missing masking noise are applied to the dataset.

8.1. Key results

Based on the conducted experiments, the key findings can be summarized as follows:

- Augmenting 80 times a nine days-long training set boosts the generalization capabilities of the existing base imputation model, to almost one year of observations.
- The average RMSEs of the augmented model are 1678.00 Wh and 1167.30 Wh, respectively for continuous and random missing scenarios. The proposed augmentation process can therefore decrease the respective RMSEs of the existing base imputation model by almost 37% and 48%.
- The augmented model outperforms all the analyzed benchmark methods. In particular, the average RMSEs are almost 23% and 12% lower than the best-performing benchmark model, respectively for continuous and random missing scenarios.
- The execution time, i.e. time required for training and validation, is approximately 10 min. The average inference time of one sample is 0.0025 s, while it is 0.077 s for 183 samples. The inference memory usage achieves a constant average value of 0.018 MiBs after the 75th inferred sample. Peaks of memory usage up to 0.086 MiBs can sporadically occur.

In summary, given only nine days of data for training, the proposed data augmentation strategy considerably boosts the performance of the existing denoising convolutional autoencoder to a completely different task. Therefore, this approach is essential to facilitate the field applications of denoising autoencoder-based imputation methods. Furthermore, the analyses related to the computational requirements reveals that the augmented model might be effectively coupled with existing real-time building control applications, as back-up option in case of sensor failure.

8.2. Limitations

Based on the conducted experiments, the limitations of the presented approach can be summarized as follows:

- The most challenging month of the year to impute in the used datasets is December, with average RMSEs of 2141.02 Wh and 1574.00 Wh, respectively for continuous and random missing scenarios.

Table A.10
Summary of the used base imputation model. Based on Liguori et al. [34].

Layer	Output shape [batch size, timesteps, features]	Parameters
Input Layer	[(None, 48, 1)]	0
Conv 1D	(None, 48, 16)	128
MaxPooling 1D	(None, 24, 16)	0
Conv 1D	(None, 24, 8)	904
MaxPooling 1D	(None, 12, 8)	0
Conv 1D	(None, 12, 8)	456
Batch Normalization	(None, 12, 8)	32
UpSampling 1D	(None, 24, 8)	0
Conv 1D	(None, 24, 16)	912
Batch Normalization	(None, 24, 16)	64
UpSampling 1D	(None, 48, 16)	0
Conv 1D	(None, 48, 1)	113
Batch Normalization	(None, 48, 1)	4
Total parameters	2613	
Trainable parameters	2563	
Non-trainable parameters	50	

- The average NRMSEs of the augmented model at the disaggregate flat level is 182% and 226% higher than at the aggregate building level, respectively for continuous and random missing scenarios.
- The average NRMSEs of the augmented model at the peak electricity consumption is 39% and 73% higher than at the continuous missing scenario, respectively for building and flat level.

In summary, the augmented model performs poorly on electricity consumption profiles characterized by higher data variability and peaks. This can be explained by considering that the existing base imputation model was optimized on much more regular and predictable daily profiles of data. Furthermore, the proposed data augmentation technique makes it possible to explore only a few combinations of missing components at the peak. In these challenging scenarios, different solutions can be adopted as follows. (1) Using a base imputation model implemented on daily profiles of data characterized by much higher variability and peaks. (2) Optimizing the architecture of the base imputation model, e.g. hyperparameter tuning. (3) Using more historical data for training. However, further experiments should be performed to prove the effectiveness of the previous points.

8.3. Future work

The following points are not addressed in detail in this paper and are recommended for future work. (1) The performance of the augmented model at the disaggregate flat level and peak electricity consumption is evaluated only for one exemplary flat from each analyzed dataset. Therefore, future work should analyze more in detail the impact of different levels of data aggregation and peak electricity consumption on the proposed method. (2) An analysis for different periodic patterns, such as weekly or monthly patterns, is not performed. As such, the augmented model can only impute sub-daily gaps of information. Future work should evaluate the presented data augmentation strategy for denoising models originally optimized on different periodic patterns. For the same reason, future work should evaluate the presented data augmentation strategy for denoising models originally optimized on different time resolutions, such as 10 min or 15 min. (3) Further studies related to buildings with different occupancy patterns, such as commercial or office buildings, and locations should be tackled in the future.

CRediT authorship contribution statement

Antonio Liguori: Conceptualization, Formal analysis, Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Romana Markovic:** Conceptualization, Writing - original draft, Writing - review & editing. **Martina Ferrando:** Conceptualization, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Jérôme Frisch:** Conceptualization, Writing - review & editing, Supervision. **Francesco Causone:** Conceptualization, Writing - review & editing, Data curation, Supervision. **Christoph van Treeck:** Conceptualization, Writing - review & editing, Funding acquisition, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Antonio Liguori reports financial support was provided by Deutsche Forschungsgemeinschaf.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – TR 892/8-1. Simulations were performed with computing resources granted by RWTH Aachen University, Germany under project rwth0622. This paper benefited greatly from discussions with members of IEA EBC Annex 79. All authors approved the version of the manuscript to be published.

Appendix

An overview of the used base imputation model from Liguori et al. [34] is presented in Table A.10. The table gives an indication of the size of the network, hence approximating the network complexity [58]. In particular, the total number of parameters is 2613.

The model takes a single feature input vector with 48 timesteps and unspecified batch size. In particular, the network is characterized by a series of one-dimensional convolutional, maxpooling, upsampling and batch normalization layers. Here, the batch normalization is used to avoid network saturation. In order to work with multi-dimensional inputs with more than one feature, the network architecture should be further optimized and multi-dimensional layers should be considered.

References

[1] Directive (EU) 2018/844 of the European parliament and of the council of 30 may 2018 amending directive 2010/31/EU on the energy performance of buildings and directive 2012/27/EU on energy efficiency. Official J Eur Union 2018;61(3):75–6, http://data.europa.eu/eli/dir/2018/844/oj.

[2] Causone F, Tatti A, Alongi A. From nearly zero energy to carbon-neutral: Case study of a hospitality building. Appl Sci 2021;11(21):10148.

[3] Hong T, Wang Z, Luo X, Zhang W. State-of-the-art on research and applications of machine learning in the building life cycle.

[4] Bengio Y, Goodfellow I, Courville A. Deep learning (Vol. 1). MIT press Massachusetts, USA; 2017.

[5] Fan C, Chen M, Wang X, Wang J, Huang B. A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Front Energy Res 2021;9:652801.

[6] Chong A, Lam KP, Xu W, Karaguzel OT, Mo Y. Imputation of missing values in building sensor data. ASHRAE IBPSA-USA SimBuild 2016;6:407–14.

[7] Xiao Z, Gang W, Yuan J, Chen Z, Li J, Wang X, et al. Impacts of data preprocessing and selection on energy consumption prediction model of HVAC systems based on deep learning. Energy Build 2022;111832.

[8] Ma J, Cheng JC, Jiang F, Chen W, Wang M, Zhai C. A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data. Energy Build 2020;216:109941.

[9] Li H, Chen X, Shan M, Duan P. Missing data filling methods of air-conditioning power consumption for public buildings. In: 2020 39th Chinese control conference. IEEE; 2020, p. 3183–7.

[10] Liu X, Zhang Z. A two-stage deep autoencoder-based missing data imputation method for wind farm SCADA data. IEEE Sens J 2021;21(9):10933–45.

[11] Hussain SN, Aziz AA, Hossen MJ, Aziz NAA, Murthy GR, Mustakim FB. A novel framework based on CNN-LSTM neural network for prediction of missing values in electricity consumption time-series datasets. J Inf Process Syst 2022;18(1):115–29.

[12] Jung S, Moon J, Park S, Rho S, Baik SW, Hwang E. Bagging ensemble of multilayer perceptrons for missing electricity consumption data imputation. Sensors 2020;20(6):1772.

[13] Ma Q, Li S, Shen L, Wang J, Wei J, Yu Z, Cottrell GW. End-to-end incomplete time-series modeling from linear memory of latent variables. IEEE Trans Cybern 2019;50(12):4908–20.

[14] Zerveas G, Jayaraman S, Patel D, Bhamidipaty A, Eickhoff C. A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021, p. 2114–24.

[15] Delasalles E, Ziat A, Denoyer L, Gallinari P. Spatio-temporal neural networks for space-time data modeling and relation discovery. Knowl Inf Syst 2019;61(3):1241–67.

[16] Li L, Zhou H, Liu H, Zhang C, Liu J. A hybrid method coupling empirical mode decomposition and a long short-term memory network to predict missing measured signal data of SHM systems. Struct Health Monit 2021;20(4):1778–93.

[17] Zhang Y-F, Thorburn PJ, Xiang W, Fitch P. SSIM—A deep learning approach for recovering missing time series sensor data. IEEE Internet Things J 2019;6(4):6618–28.

[18] Flores A, Tito-Chura H, Yana-Mamani V. Wind speed time series imputation with a bidirectional gated recurrent unit (GRU) model. In: Proceedings of the future technologies conference. Springer; 2021, p. 445–58.

[19] Chen Z, Yuan C, Wu H, Zhang L, Li K, Xue X, et al. An improved method based on EEMD-LSTM to predict missing measured data of structural sensors. Appl Sci 2022;12(18):9027.

[20] Flores A, Tito H, Centty D. Recurrent neural networks for meteorological time series imputation. Int J Adv Comput Sci Appl 2020;11(3).

[21] Zhou Y, Jiang J, Yang S-H, He L, Ding Y. MuSDRI: Multi-seasonal decomposition based recurrent imputation for time series. IEEE Sens J 2021;21(20):23213–23.

[22] Zhang W, Zhang P, Yu Y, Li X, Biancardo SA, Zhang J. Missing data repairs for traffic flow with self-attention generative adversarial imputation net. IEEE Trans Intell Transp Syst 2021.

[23] Qian F, Gao W, Yang Y, Yu D, et al. Potential analysis of the transfer learning model in short and medium-term forecasting of building HVAC energy consumption. Energy 2020;193(C).

[24] Siłka J, Wieczorek M, Woźniak M. Recurrent neural network model for high-speed train vibration prediction from time series. Neural Comput Appl 2022;1–14.

[25] Kreuzer D, Munz M. Deep convolutional and LSTM networks on multi-channel time series data for gait phase recognition. Sensors 2021;21(3):789.

[26] He X, Zhao K, Chu X. Automl: A survey of the state-of-the-art. Knowledge-based systems. 2021.

[27] Dong B, Markovic R, Carlucci S, Liu Y, Wagner A, Liguori A, et al. A guideline to document occupant behavior models for advanced building controls. Build Environ 2022;109195.

[28] Guideline A. Guideline 13-2015, Specifying building automation systems. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. Atlanta, GA; 2000.

[29] Jeong D, Park C, Ko YM. Missing data imputation using mixture factor analysis for building electric load data. Appl Energy 2021;304:117655.

[30] Fan C, Xiao F, Zhao Y, Wang J. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. Appl Energy 2018;211:1123–35.

[31] Liu Y, Pang Z, Karlsson M, Gong S. Anomaly detection based on machine learning in IoT-based vertical plant wall for indoor climate control. Build Environ 2020;183:107212.

[32] Araya DB, Grolinger K, ElYamany HF, Capretz MA, Bitsuamlak G. Collective contextual anomaly detection framework for smart buildings. In: 2016 International joint conference on neural networks. IEEE; 2016.

[33] Loy-Benitez J, Li Q, Nam K, Yoo C. Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation. Sustainable Cities Soc 2020;52:101847.

[34] Liguori A, Markovic R, Dam TTH, Frisch J, van Treeck C, Causone F. Indoor environment data time-series reconstruction using autoencoder neural networks. Build Environ 2021;191:107623.

[35] Liguori A, Markovic R, Frisch J, Wagner A, Causone F, van Treeck C. A gap-filling method for room temperature data based on autoencoder neural networks. In: Building simulation conference 2021: 17th conference of IBPSA. IBPSA; 2021.

[36] Pinto G, Wang Z, Roy A, Hong T, Capozzoli A. Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. Adv Appl Energy 2022;100084.

[37] Fan C, Chen M, Tang R, Wang J. A novel deep generative modeling-based data augmentation strategy for improving short-term building energy predictions. In: Build Simul. 15, (2):Springer; 2022, p. 197–211.

[38] Wu D, Hur K, Xiao Z. A GAN-enhanced ensemble model for energy consumption forecasting in large commercial buildings. IEEE Access 2021;9:158820–30.

[39] Lu Y, Tian Z, Zhang Q, Zhou R, Chu C. Data augmentation strategy for short-term heating load prediction model of residential building. Energy 2021;235:121328.

[40] Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans Image Process 2006;15(12):3736–45.

[41] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11(12).

[42] Arslan M, Guzel M, Demirci M, Ozdemir S. SMOTE and Gaussian noise based sensor data augmentation. In: 2019 4th International conference on computer science and engineering. IEEE; 2019, p. 1–5.

[43] Freund RJ, Wilson WJ, Mohr DL. CHAPTER 2 - probability and sampling distributions. In: Freund RJ, Wilson WJ, Mohr DL, editors. Statistical methods (Third Edition). third ed.. Boston: Academic Press; 2010, p. 67–124. http://dx.doi.org/10.1016/B978-0-12-374970-3.00002-0, URL https://www.sciencedirect.com/science/article/pii/B9780123749703000020.

[44] Bishop CM. Regularization and complexity control in feed-forward networks. 1995.

[45] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature 2020;585(7825):357–62. http://dx.doi.org/10.1038/s41586-020-2649-2.

[46] Lillstrang M, Harju M, del Campo G, Calderon G, Röning J, Tamminen S. Implications of properties and quality of indoor sensor data for building machine learning applications: Two case studies in smart campuses. Build Environ 2022;207:108529.

[47] Simmini F, Agostini M, Coppo M, Caldognetto T, Cervi A, Lain F, et al. Leveraging demand flexibility by exploiting prosumer response to price signals in microgrids. Energies 2020;13(12):3078.

[48] Basu K, Debusschere V, Douzal-Chouakria A, Bacha S. Time series distance-based methods for non-intrusive load monitoring in residential buildings. Energy Build 2015;96:109–17.

[49] Wannesm, khendrickx, Yurtman A, Robberechts P, Vohl D, Ma E, et al. Wannesm/dtaidistance: v2.3.5. 2022, http://dx.doi.org/10.5281/zenodo.5901139.

[50] Raubitzek S, Neubauer T. A fractal interpolation approach to improve neural network predictions for difficult time series data. Expert Syst Appl 2021;169:114474.

[51] Stekhoven DJ, Bühlmann P. MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics 2012;28(1):112–8.

[52] MissingPy. 2015, https://github.com/epsilon-machine/missingpy. [Accessed 06 October 2022].

[53] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, et al. Missing value estimation methods for DNA microarrays. Bioinformatics 2001;17(6):520–5.

[54] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[55] Lotfi M, Javadi M, Osório GJ, Monteiro C, Catalão JP. A novel ensemble algorithm for solar power forecasting based on kernel density estimation. Energies 2020;13(1):216.

[56] Liguori A, Yang S, Markovic R, Dam TTH, Frisch J, Wagner A, et al. Prediction of HVAC loads at different spatial resolutions and buildings using deep learning models. In: Building simulation conference 2021: 17th conference of IBPSA. IBPSA; 2021.

[57] RWTH compute cluster. 2022, URL https://www.itc.rwth-aachen.de/cms/IT-Center/Forschung-Projekte/High-Performance-Computing/~eucm/Infrastruktur/. [Accessed 05 June 2022].

[58] Hu X, Chu L, Pei J, Liu W, Bian J. Model complexity of deep learning: A survey. Knowl Inf Syst 2021;63(10):2585–619.