Indoor environment data time-series reconstruction using autoencoder neural networks

Antonio Liguori [a,*], Romana Markovic [b], Thi Thu Ha Dam [a], Jérôme Frisch [a], Christoph van Treeck [a], Francesco Causone [c]

[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

ABSTRACT

As the number of installed meters in buildings increases, there is a growing number of data time-series that could be used to develop data-driven models to support and optimize building operation. However, building data sets are often characterized by errors and missing values, which are considered, by the recent research, among the main limiting factors on the performance of the proposed models. Motivated by the need to address the problem of missing data in building operation, this work presents a data-driven approach to fill these gaps. In this study, three different autoencoder neural networks are trained to reconstruct missing short-term indoor environment data time-series in a data set collected in an office building in Aachen, Germany. This consisted of a four year-long monitoring campaign in and between the years 2014 and 2017, of 84 different rooms. The models are applicable for different time-series obtained from room automation, such as indoor air temperature, relative humidity and CO_2 data streams. The results prove that the proposed methods outperform classic numerical approaches and they result in reconstructing the corresponding variables with average RMSEs of 0.42 °C, 1.30 % and 78.41 ppm, respectively.

1. Introduction

In the European Union buildings account for more than 40 % of the total final energy consumption and approximately 36 % of CO_2 emissions [1]. As a consequence, reliable estimation of building consumption data could foster energy efficiency strategies, such as the analyses of retrofit options [2] or the development of fault detection and diagnosis (FDD) schemes [3]. In the related research, two approaches are generally followed to achieve this goal [4]: forward modeling and data-driven modeling. While the former is based on solid engineering principles, the latter relies on data collected during normal or predetermined system operation and it can usually capture more accurate as-built system's performance with a limited number of known parameters [4]. Additionally, data-driven approaches can be successfully applied to represent energy-related human actions in buildings (e.g. window openings), being the result of a number of stochastic driving forces [5,6].

By definition, data-driven modeling explicitly requires the availability of useful data [7]. Therefore, missing values present the major limitation on this approach [8]. As stated by multiple studies [7,8], data gaps are a common problem in building automation systems (BAS) and they may be caused by a number of reasons such as power outages, sensors defects, communication problems or network issues. As a result, the presence of these anomalies could significantly reduce the size of the available data set and hinder further energy analysis [9,10]. So far, existing studies have often handled missing data either using simplified methods [8] or excluding them from further analytics due to the lack of ground truth values [7]. In summary, both latter approaches have usually led to limited inserting accuracy and lower resulting model performance [7,8].

Even though the related research has already identified some promising techniques to address missing values and anomalies in monitoring building data sets [11–17], some significant research questions remain still unaddressed. In particular, the existing studies have often either focused on reconstructing a single type of signal or made use of a number of specific information from additional sensors. Furthermore, the advance of deep learning in the building related research [18–37]

---

opens to more interesting possibilities that have not yet been fully explored, due to the small amount of available data and computational power experienced in most of the existing studies.

Based on the previous considerations, the aim of this paper is to propose a deep learning-based method for reconstructing missing sequences of various indoor environment streams obtained from room control sensors. Due to the frequency of short-term gaps (i.e. < 24 h) in building data sets [38,39], the focus is on the inserting of small intervals of values. For that purpose, based on data availability to the authors, a data set collected in an office building located in Aachen, Germany, was analyzed and preprocessed. This consisted of a four year-long monitoring campaign of 84 different rooms and it could, therefore, provide significant generalization capability to the implemented approach. Models for handling missing data points were implemented, trained and evaluated on indoor air temperature (T), relative humidity (RH) and CO_2 concentration data. In particular, due to the increasing use of autoencoder neural networks in the FDD-related research for buildings' control [40–44], three promising autoencoder architectures were investigated: feed-forward denoising autoencoder, convolutional denoising autoencoder and long short-term memory (LSTM) denoising autoencoder. The optimal problem hypothesis was identified based on the results obtained from 7,000 core hours of GPU and CPU computations. Eventually, the performance of the proposed methods were compared to analytical methods based on polynomial interpolations.

The scientific contribution of the presented work consists of the following:

- To explore the full potential of autoencoder neural networks for reconstructing indoor environment data time-series.
- To analyze the performance variability of deep learning models when applied to different kinds of monitoring building data.
- To present a generalized gap-filling method to address the problem of missing values in building data sets.
- To propose a solution to address the issue of artificial neural networks' (ANNs) saturation for energy systems applications.

The rest of this paper is organized as follows: Section 2 presents the motivation that led to the development of a missing data inserting model based on autoencoder neural networks. Section 3 provides the reader with further information on the used data set and on the models' theory and implementation. Section 4 presents results on developing a suitable tool for indoor environment data time-series reconstruction. Finally, the results and novel findings are discussed and summarized in Sections 5 and 6. A summary of the used abbreviations is available in Table 1.

2. Background

2.1. Missing data in buildings' control

The importance of sufficient large data sets on time-series modeling was empirically explored in the scope of the Texas LoanSTAR program [9], whose objective was to measure savings from energy conservation retrofits. By increasing the length of building data sets from one to five months, the average cooling prediction error decreased from 7.3% to 3.0% and the annual heating prediction error decreased from 27.5% to 12.9%. Zapata et al. [10] discovered also that a fast fourier transform (FFT), as applied to assess the frequency content of time-series wind speed data, gave unacceptable results when information loss was at least 2.5% of the data set. Therefore, the presence of missing values in building data sets may affect the performance of the adopted method and hinder further energy analysis. In particular, the analysis of an energy use data set, collected in over 200 buildings in Texas [9], revealed that about 60 % of the observed missing data were in the range between 1 and 6 h [38,39].

As pointed out by Chong et al. [8], in 2016 there was still little relevant research about handling missing values in building data sets.

Table 1
List of abbreviations.

ANNs	Artificial neural networks
BAS	Building automation systems
BIT	Bi-directional imputation and transfer learning
BN	Bayesian network
CR	Corruption rate
DBN	Deep belief network
ELM	Extreme learning machine
EM	Expectation–maximization
FDD	Fault detection and diagnosis
FFT	Fast fourier transform
GANs	Generative adversarial networks
IAQ	Indoor air quality
IQR	Interquartile range
LSTM	Long short-term memory
MAE	Mean absolute error
MELs	Miscellaneous electric loads
MSE	Mean squared error
NRMSE	Normalized root mean squared error
OB	Occupant behavior
OWA	Optimally weighted average
RBMs	Restricted boltzmann machines
RF	Random forest
RH	Indoor relative humidity
RMSE	Root mean squared error
RNN	Recurrent neural network
SAT	Saturation performance metric
SGD	Stochastic gradient descendent
T	Indoor air temperature

As a consequence, existing studies often relied on simplified methods such as complete case analysis, mean inserting and zeros inserting [8]. However, these methods often resulted in poor reconstruction of missing data and they often led to limited performance of later applied data-driven models. In recent years, the topic of missing data inserting for building monitoring data has gained increasing importance and different studies have been carried out. Candanedo et al. [11] reconstructed the average indoor air temperatures of a passive house, achieving accurate results with a random forest (RF) model. Rahman et al. [12] presented a recurrent neural network (RNN) for the prediction of electricity consumption in commercial and residential buildings. The same model was applied for the missing values inserting in the event of large data gaps (i.e. bigger than 5 h). However, the models proposed in the previous studies used as input features multiple time-series (e.g. external weather data, total electrical energy use, indoor variables) that could not be always known for a data reconstruction problem. In this regard, a further improvement was noted in Benitez et al. [13] and Wang et al. [14]. Benitez et al. [13] coupled multivariate variational autoencoders with convolutional layers to estimate missing indoor air quality (IAQ) subway data. The model's inputs consisted of 9 different IAQ variables and the missing values were reconstructed based on the captured correlations between these variables. A similar approach was followed in Wang et al. [14]. Here, a FDD method based on expectation–maximization (EM) algorithm and Bayesian network (BN) was developed to insert missing values from a chiller's data set. In this case, the model's inputs consisted of 9 different variables that indicated the healthy status of the system. Hence, in the last two approaches, weather information were not exploited but measurements from multiple sensors were required. This may represent a particular complexity for the models, since the required meters' data might be not available for diverse reasons. Differently from the previous studies, the models proposed in Peppanen et al. [15] and Ma et al. [16] had the advantage to need only the historical measurements of the same type of data (i.e. power), without additional specific information. Peppanen et al. [15] implemented an optimally weighted average (OWA) load power imputation method for smart meter measurements. Ma et al. [16] applied an univariate LSTM with bi-directional imputation and transfer learning (BIT) for electric consumption data inserting of a campus lab building, achieving a reconstruction error approximately

30 % less than linear interpolation models. However, since these studies focused only on the reconstruction of a single signal, it is not clear how the proposed models would behave when applied to other types of building monitoring data.

In summary, there is a gap in the related research that does not address the missing data reconstruction performance variability of a single model when applied to different single streams of building monitoring data. Few works are found to analyze this case and they either focus on simplified approaches (i.e. forward propagation) [17] or on advanced methods without further optimization [17]. This paper tries to fill this gap, by exploiting the full potential of deep learning on diverse indoor environment streams.

2.2. Deep learning methods for buildings' control

Deep learning models are neural networks with learned feature representation over multiple hidden layers [45]. In the related building research, these methods have been extensively used for energy consumption and occupant behavior (OB) modeling applications [18–37]. Qian et al. [27] explored the potential of ANNs for HVAC load forecasting, when applied to small amount of data. The results proved that the fitting degree of the proposed models was over 85 %. They pointed out the importance of the sufficiently large training data set. In particular, the use of a smaller training set that consisted of one month and one week of monitoring data led to accuracy decrease for 6 % and 20 %, respectively. Zhang et al. [19] trained a deep belief network (DBN) and extreme learning machine (ELM) based framework to predict half-hourly building energy consumption data. Here, DBNs consisted of a stack of restricted boltzmann machines (RBMs), where RBMs had fully connected visible and hidden layers [19]. The correspondent mean absolute error (MAE) was 10 % higher than the results obtained by support vector regression. Yan et al. [30] addressed the issue of the imbalanced properties of training data sets for automatic FDD of chillers. They used generative adversarial networks (GANs) to generate faulty training samples. The results proved that without the implemented model, classification accuracy could hardly reach 90 %. Conclusively, Markovic et al. [36] developed a LSTM neural network for day-ahead prediction of miscellaneous electric loads (MELs). The proposed implementation outperformed benchmark approaches based on Weibull distribution and Gaussian mixture methods when MELs and occupancy information data were used as input parameters to the model.

2.3. Autoencoder neural networks for buildings' control

Autoencoders are ANNs which learn to reconstruct original inputs from a noisy version, making missing data reconstruction one of their reasonable applications [46]. In the field of building energy systems, these models have been increasingly applied for FDD purposes [40–44]. Fan et al. [40] explored the potential of different types of autoencoder neural networks in the anomaly detection of building operational data. The results showed that a 1D convolutional architecture could effectively capture the intrinsic characteristics in building energy data, while preserving the temporal data information. Liu et al. [41] applied different machine learning-based anomaly detection methods to vertical plant wall systems. The results confirmed that the autoencoder configuration outperformed other models for both contextual and point anomaly detection of temperature and CO_2 data. Araya et al. [42] used autoencoders to capture HVAC consumption patterns and they used the gained knowledge to identify abnormal consumption behaviors in the same system. Legrand et al. [43] applied different autoencoder architectures for the anomaly detection of 19 sensors in multiple houses. The results showed that RNN autoencoders had better generalization capabilities than the other used configurations. Benitez et al. [44] developed an autoencoder-based sensor validation method for the FDD of a subway system. This approach outperformed conventional methods for detecting and reconstructing faulty measurements.
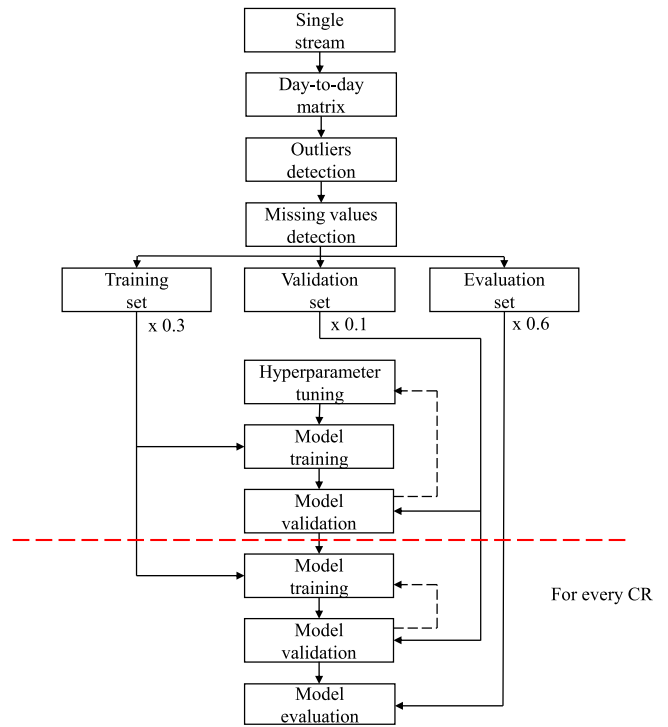

Fig. 1. Modeling flowchart. CR is the applied masking noise.

3. Methodology

The aim of this paper is to develop an approach for filling short-term indoor environment data gaps. For that purpose, three different autoencoder neural network architectures were implemented in order to identify the optimal model hypothesis. As described in Fan et al. [40], before further model development, it is common to rearrange the time-series into a matrix of sub-sequences. The length of each sub-sequence is selected based on the most dominant period identified in the data. In this work, indoor air temperature, relative humidity and CO_2 concentration data were found to have clear daily seasonality. Therefore, the indoor environment streams were rearranged into day-to-day matrices. Missing data in each stream were then simulated by setting to zero sub-daily sequences of random length. In particular, the reconstructed gaps ranged between few hours (10 % of the daily values) and around 22 h (90 % of the daily values). A summary of the adopted modeling approach for each indoor environment stream is proposed in Fig. 1.

3.1. Denoising autoencoder neural networks

The general structure of an autoencoder is presented in Fig. 2. An autoencoder neural network is a representation learning approach which turns incoming data into different representations, through an encoder function, and reconstructs the original input through a decoder function [45].

Input x is mapped to an output reconstruction r through an internal representation h, namely code. The autoencoder has two components: the encoder f (mapping x to h) and the decoder g (mapping h to r) [45].

The encoder is defined as [45]:

$$h = c(Wx + b),\tag{1}$$

while the decoder is formulated as:
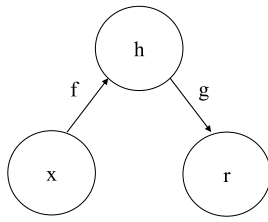
$$r = c(W'h + b'),\tag{2}$$

Fig. 2. Working principle of a general autoencoder.
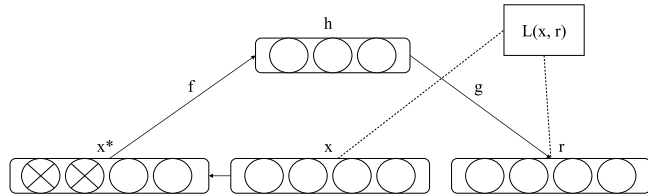*Source:* Figure reproduced based on Goodfellow [45].

Fig. 3. Working principle of a denoising autoencoder.
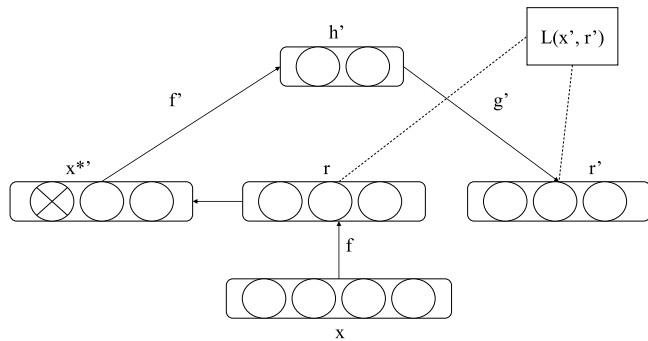*Source:* Figure reproduced based on Vincent et al. [47].

Fig. 4. Working principle of a stacked denoising autoencoder.
Source: Figure reproduced based on Vincent et al. [47].

where c is a non-linear function, called activation function, W and W' are called weights, b and b' are called biases. By implementing activation several times (training), the model could acquire useful knowledge about the systems' properties [45].

The structure of a denoising autoencoder is presented in Fig. 3, while its extension to the stacked denoising autoencoder is presented in Fig. 4. In contrast to a general autoencoder, a denoising autoencoder receives a corrupted input x^* and is trained to reconstruct the original uncorrupted data x [45].

The training process consists of minimizing a loss function $L(x, r)$ which quantifies the difference between the original input and output at each step. In this study, corruption of input data is performed by setting to zero an interval of sequential values of random length. This approach is used to simulate how missing data are distributed.

After the first level of denoising autoencoder has been trained (Fig. 3), a second level of denoising autoencoder is trained using the previously optimized encoding function, f. Corruption takes place on the output of the previous optimized layer r [47].

3.2. Data set

The used data were collected in the E.ON ERC main building, located in Aachen, Germany. The building under investigation is a mechanically ventilated building with a usable area of 7,500 m² over four storeys. The same building has U-values of 1.22 $\frac{W}{m^2K}$ for windows

and 0.19 $\frac{W}{m^2K}$ for the outer walls, a yearly primary energy demand of 105.1 $\frac{kWh}{m^2a}$ and a net energy demand of 68 $\frac{kWh}{m^2a}$ [48–51].

Temperature, relative humidity and CO_2 concentration data were recorded in every room of the building, namely offices, seminar and conference rooms, staff facilities, laboratories and common area. Here, each sensor was integrated within a BACnet network. Room temperature was measured by means of thermocouples, which were calibrated by measuring and comparing the temperature offsets with the actual sensors' accuracy. Due to data sharing agreement regarding the hardware installation, no additional information about the RH and CO_2 sensors' architecture and calibration was provided to the authors in the energy report of the building.

Based on the logging frequency and monitoring duration, it could be expected that around 181 million sets of observations were collected for each variable from 2014 to 2017. The monitoring data were grouped in two subsets, namely data set "A" (2014–2015) and "B" (2016–2017). Data set "A" contained measurements for 73 rooms from 2014 and 2015, stored into "HDF5" data containers [52] on a monthly basis. Monitoring data for the years 2016 and 2017 were collected in 84 offices and they were stored in "pickle" files [53]. Here, each file contained data for a single office over the whole observed biannual period.

3.3. Data preprocessing

Before further analysis and modeling, data were cleaned and preprocessed. This step involved the detection of frequently encountered anomalies, such as missing values and outliers [54]. Missing values reduce the size of the available data set, hence compromising the reliability of the model's outcome [54]. Outliers are noisy data points with values significantly different from the majority of other data points [54]. For this reason, they could lead to underestimated or overestimated results [54]. For the detailed explanation of the adopted data cleaning procedure the reader is referred to Appendix A.1.

According to the existing literature [55–57], additional preprocessing, such as resampling resolution and data normalization was performed in order to increase performance and computational efficiency of the proposed models. Here, resampling resolution is the process of changing the frequency of time-series data [55], while data normalization is applied to prepare raw data for a better network use [56]. In particular, data were downsampled to 30 min frequency and normalized using the Statistical or Z-Score Normalization function [56]:

$$z = (x - u)/s , \tag{3}$$

where, x are data to normalize, u is the mean of the training samples and s is the standard deviation of the training samples. Before normalizing, data were split into training (30 %), validation (10 %) and evaluation set (60 %) for every variable. In order to favor models' generalization, limits were defined based on training set and eventually adopted for each of the three parts identically.

3.4. Model development

The models were developed using the Python programming language and open source libraries Tensorflow [58] and Keras [59]. Fig. 5 shows the autoencoder structure as implemented in the scope of the performed experiments. The explored models included the architectures with one and up to three hidden layers per each encoder and decoder. Batch-normalization was included after every layer in order to avoid the network saturation for both feed-forward and convolutional autoencoders [60].

Feed-forward denoising autoencoders were fed with unrolled half-hourly daily observations, which resulted in 48 features. Unrolling the 1-D temporal sequences into a single input layer is, indeed, a commonly adopted approach to address temporal dependencies for a feed-forward
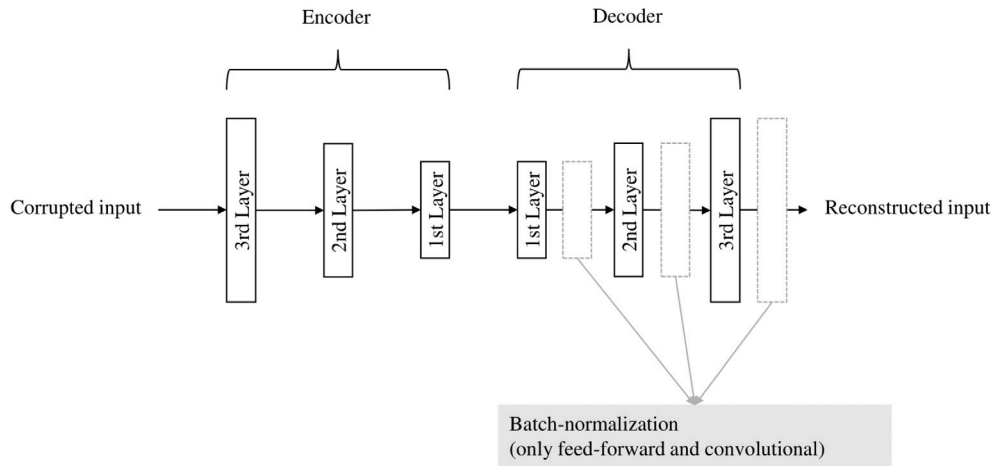
Fig. 5. General autoencoder architecture with 3 layers per side.

Table 2
Overview of tuned hyperparameter and explored values.

Hyperparameter	Values
Hidden layers per side	1–3
Hidden layer units/filters (convolutional)	8, 16, 32, 64, 128
Kernel size (convolutional)	2, 3, 5, 7, 11
Batch size	128
Learning rate	0.001, 0.01, 0.1
Optimizer	SGD with 0.9 momentum, Adam

neural network [5,37]. However, feed-forward neural networks with unrolled sequences have separate parameters for each feature, which means that weights cannot be shared along the input series [45]. While convolutional neural networks apply the same kernel to every time-step, in the LSTM configuration each output node is a function of the previous one and parameters can be shared along very long sequences [45].

As common for the supervised learning, the overfitting can occur when the number of training iterations is set too high [61]. In that case, the training error keeps decreasing, while the validation error increases due to lower model's generalization capabilities [45]. The similar phenomena could be observed in case of this study and occurrence of overfitting is exemplary presented in Appendix A.2. Therefore, the early stopping criteria was introduced in order to avoid overfitting [45]. Furthermore, the mean squared error (MSE) loss function was applied to the reconstructed and original input over all the training samples, as follows [45]:

$$MSE = \frac{\sum_i^m (Y' - Y)_i^2}{m} ,\tag{4}$$

where m is the batch size. In order to find the optimal parameters for the models, the MSE was minimized using either a stochastic gradient descendent (SGD) optimizer with momentum [62] or Adam optimizer [63]. The optimizer choice was handled as an additional hyperparameter. In this study, the hyperparameter tuning was conducted separately for each target variable (temperature, relative humidity, CO_2 concentration). Additionally, the whole hyperparameter range was explored for all neural units, such as feed-forward, LSTM and convolutional. The range of values in which the optimal model's configuration was investigated is summarized in Table 2.

In the scope of experimental design setting, both grid search and random search for hyperparameter tuning were considered. Eventually, it was opted for grid search, due to the relative small hyperparameter space. Namely, the searched hyperparameter space consisted of approximately 1,890 model hypotheses, and no significant computational efficiency would be achieved by opting for an alternative

hyperparameter tuning strategy. However, in case of the larger hyperparameter space, alternative tuning strategies should be considered, such as random or greedy search [64].

For every possible combination of hyperparameter, the score given to the model was obtained by averaging the MSEs, computed on the normalized validation set, with a corruption rate (CR) ranging between 0.2 and 0.8. Given the computational cost of this process, multiple independent jobs with different hyperparameters were run in parallel using the computational resources at the RWTH Aachen University Compute Cluster. In order to address the stochastic initialization of models' weights, the tuned configurations were run again 10 times and evaluated on the same validation data as before. Models with the lowest MSE were eventually exported for further evaluation on the evaluation set.

4. Results

4.1. Performance evaluation metrics

The performance of missing data insertion was assessed using the root mean squared error (RMSE) method, since it is the established evaluation method by the existing research [11,13,16]. In order to obtain objective evaluation results, the RMSE was applied to all the corrupted evaluation data. Additionally, the ability of the proposed models for capturing indoor environmental data patterns was estimated by computing the RMSE on each sequence [40]. The RMSE equation is given as follows [11,16,40]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_i^{obs} - X_i^{inserted})^2}{n}} ,\tag{5}$$

where X_i^{obs} are the ith real values, $X_i^{inserted}$ are the ith reconstructed values and n are the total number of data points on which RMSE is computed.

Comparison between different variables and studies was made by means of the normalized root mean squared error (NRMSE), obtained by normalizing the RMSE over the interquartile range (IQR), due to the possible presence of noisy data points:

$$NRMSE = \frac{RMSE}{IQR} .\tag{6}$$

4.2. General observations during model development

Network saturation was identified to be a major modeling complexity in case of both feed-forward and convolutional denoising autoencoders. Formally, neural network saturation could be described as an impediment to gradient propagation [36,65]. Its main effect was to

produce always the same output, no matter how different the input sequence was.

As presented earlier, neural networks are trained by minimizing a loss function between targets and true values. This can be accomplished by applying an iterative algorithm called gradient descendent [66]. The working principle of a gradient descendent algorithm is simple: weights and biases are initialized to some values and then they are continuously updated in the direction that decreases the loss function, namely opposite to the gradient [66]. Each unit in a neural network receives signals and weights from previous units and computes a value called pre-activation [66]:

$$z = \sum_j w_j x_j + b , \qquad (7)$$

where z is the pre-activation value, j is the number of input signals and weights, w are the weights, b is the bias which determines units' activation in case no inputs are present. The activation value is the pre-activation passed through an activation function ϕ [66]:

$$a = \phi(z) . \qquad (8)$$

If, during training, the activation of a neural network unit is always near the boundaries of its dynamic range (the possible outputs of an activation function), then the gradient of the pre-activation is very small and weights are not updated [66]. These neural network units are called saturated units [66]. Hence, saturated units can be identified looking at the histogram of the average activations and checking that they are not concentrated at the endpoints [66]. Different approaches are followed in the literature to avoid network saturation. Glorot and Bengio [65] proposed a novel normalized initialization for neural network weights:

$$W = U[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}] , \qquad (9)$$

where W are the neural network weights, U is a uniform distribution function, n is the size of the jth layer. Maas et al. [67] used ReLU activation functions as alternative to sigmoid and $tanh$, causing in this way network saturation at exactly 0. Ioffe and Szegedy [60] proposed a new mechanism called batch-normalization, which ensures that distribution of nonlinearity inputs (i.e. pre-activations) remains more stable as the network trains.

In this study, since no other formal metrics were available at the current state of the research, network saturation was evaluated using the saturation performance metric (SAT) proposed by Markovic et al. [36], calculated over the normalized reconstructed values.

In order to guarantee an effective model training, all the previous approaches followed in the literature were implemented [60,65,67]. Fig. 6 shows the histogram of the average activations for the original feed-forward denoising autoencoder configuration with classical uniform weights initialization, sigmoid activation functions and no batch-normalization. It is possible to observe the typical network saturation problem, being the average activations always concentrated at 0 and 1, namely the endpoints of a sigmoid activation function.

The use of normalized initialization, ReLU activation function and batch-normalization as proposed in the literature confirmed to reach higher performance and to overcome saturation problems. Fig. 7 shows how average network activations change during training, not being stuck in any saturating point. Note that ReLU activation function was applied on the encoder layer while batch-normalization only in the decoder layer. The authors realized that higher performance could be reached using a $tanh$ activation function in the decoder layer, in association with batch-normalization. Conclusively, the SAT results confirmed the superiority of the chosen models' configurations with respect to the original denoising autoencoder architectures with classical uniform weights initialization, sigmoid activation functions and no batch-normalization.

Table 3
SAT for denoising autoencoder neural networks for different CR. "CONV", "FEED", "LSTM" and "Avg" stand for convolutional, feed-forward, LSTM denoising autoencoder and average.

	CR [-]	T [-]			RH [-]			CO_2 [-]		
		CONV	FEED	LSTM	CONV	FEED	LSTM	CONV	FEED	LSTM
SAT	0.10	0.88	0.85	0.84	0.95	0.97	0.98	0.78	0.83	0.77
	0.20	0.87	0.83	0.79	0.95	0.98	0.98	0.71	0.77	0.76
	0.30	0.83	0.82	0.77	1.00	0.98	0.98	0.65	0.73	0.69
	0.40	0.77	0.75	0.73	0.99	0.96	0.99	0.75	0.66	0.59
	0.50	0.73	0.72	0.72	0.98	0.97	0.98	0.57	0.60	0.47
	0.60	0.73	0.68	0.65	0.97	0.95	0.95	0.52	0.46	0.35
	0.70	0.69	0.69	0.59	0.94	0.96	0.96	0.51	0.48	0.32
	0.80	0.64	0.64	0.60	0.92	0.94	0.94	0.44	0.41	0.30
	0.90	0.62	0.63	0.57	0.93	0.96	0.93	0.39	0.34	0.31
	Avg	0.75	0.73	0.70	0.96	0.96	0.97	0.59	0.59	0.51

The saturation performance results of the proposed models for different CR are summarized in Table 3. Here, network saturation is defined as the SAT lower than 0.1 [36]. Every autoencoder was well above the previous defined limit, meaning that the adopted architecture strategies could efficiently overcome saturation issues. The above identified metric was function of the particular variable and of the applied CR. While the SAT was insensitive to changes in CR for relative humidity, it decreased for both temperature and CO_2 data.

4.3. Data reconstruction performance evaluation

Firstly, the ability of autoencoder neural networks for capturing daily patterns of environmental data is assessed. Eventually, the proposed models' performance in reconstructing sub-daily indoor environment data gaps are evaluated and compared to classic polynomial interpolations.

Fig. 8 shows the RMSE variance for a LSTM denoising autoencoder. In particular, the model was trained with a masking noise of 0.1 and applied to the non corrupted evaluation sets. Here, the RMSE was computed on the single sequence as seen in Section 4.1. The average RMSEs over the number of evaluation sequences were 0.027 °C, 0.12 % and 4.25 ppm, respectively for T, RH and CO_2 data. Furthermore, all the reconstruction residuals computed with the LSTM architecture were lower than the convolutional and feed-forward ones.

Based on the boxplots presented in Fig. 8, the observed days with RMSE out of the measured IQR are represented. These could be considered as sequences with atypical behavior, for which the indoor environment data patterns cannot be detected by the model. Fig. 8 shows also, for every variable, how a random day with atypical behavior looks like and how it is reconstructed by the LSTM autoencoder. In particular, the represented sequence of relative humidity data presents an outlier in the early morning observation. This anomaly, which could be caused by sensors' malfunctioning, was not detected during the data preprocessing, but it was identified with the proposed model.

Table 4 summarizes the behavior of all the models, at different CR, when applied to the corrupted data. For every variable there was a polynomial degree for which the daily data trends were better fit by the interpolation. While indoor air temperature data were more accurately described by cubic correlations, relative humidity and CO_2 concentration data had, respectively, more linear and quadratic trends. However, all autoencoders performed by a large margin better than baseline approaches for all variables. In particular, the performance of the convolutional configuration outperformed, in average, all the alternative models. In this regard, the RMSE was 37 % lower than cubic interpolation for indoor air temperature, 24 % lower than linear interpolation for relative humidity and 30 % lower than quadratic interpolation for CO_2 concentration. In terms of NRMSE, missing relative humidity data could be reconstructed with higher accuracy, when compared to other variables. On that place, the worst behavior was
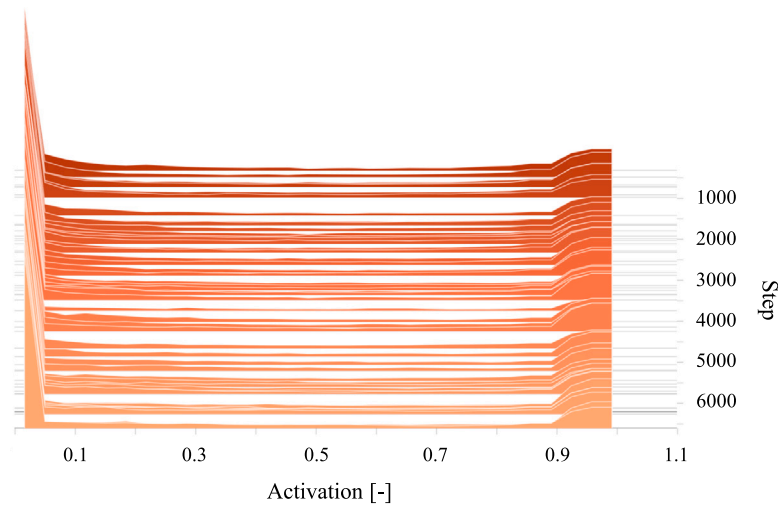
Fig. 6. 3D Histogram of the average activations for the original feed-forward denoising autoencoder with 10 % corruption rate.
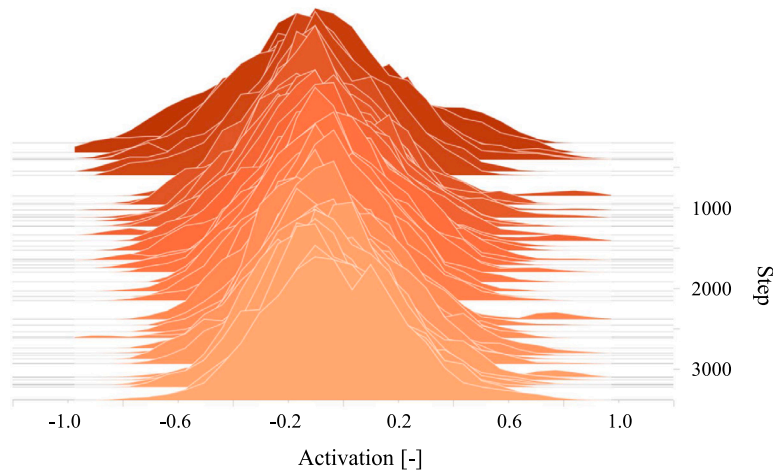

Fig. 7. 3D Histogram of the average activations for the new feed-forward denoising autoencoder with 10 % corruption rate.
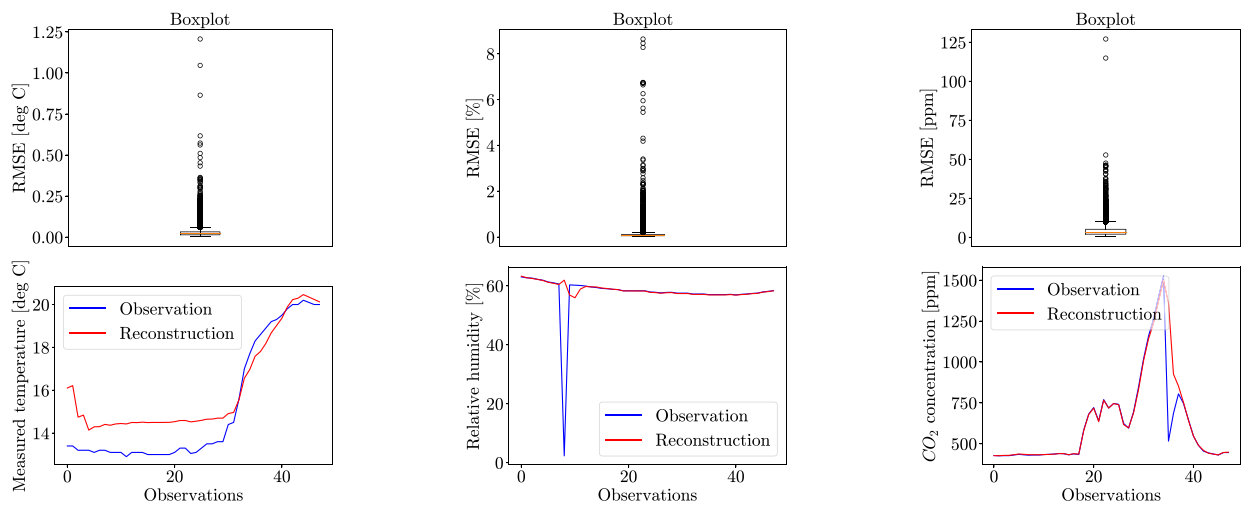

Fig. 8. Boxplots of the observed sequences and reconstruction of a day with atypical behavior.

obtained with CO_2 data. In particular, the NRMSE of the convolutional configuration in case of *RH* data was 75 % lower than *T* and 90 % lower than CO_2. These results were consistent with the SAT trend, being higher for *RH* and lower for CO_2 (Table 3).

Fig. 9 shows the density distribution of the reconstruction residuals produced by the best models identified by the previous table (i.e. convolutional autoencoder), for a corruption rate of 0.5. Here, the reconstruction residuals were computed as the differences between
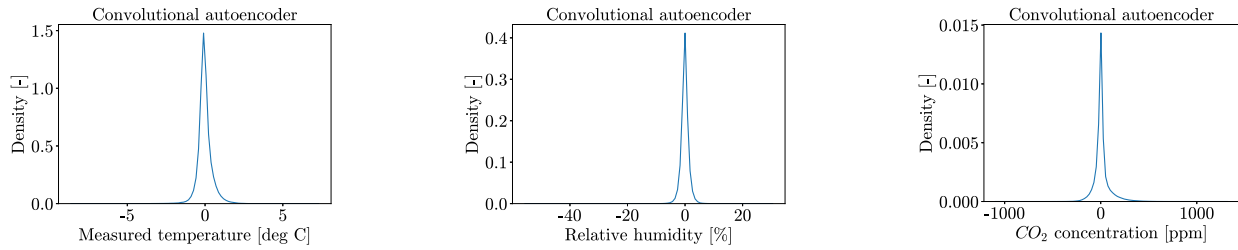
Fig. 9. Density distribution of the reconstruction residuals for a corruption rate of 0.5.

Table 4
Performance of denoising autoencoder neural networks and polynomial interpolations for reconstructing sub-daily indoor environment data gaps. "CONV", "FEED", "LSTM" stand for convolutional, feed-forward and LSTM denoising autoencoder. "LIN", "QUAD" and "CUB" stand for linear, quadratic and cubic interpolation. "Avg" stands for average.

	CR [-]	T [°C]			RH [%]			CO_2 [ppm]		
		CONV	FEED	LSTM	CONV	FEED	LSTM	CONV	FEED	LSTM
RMSE	0.10	0.22	0.32	0.33	0.73	1.14	1.05	49.10	64.58	64.88
	0.20	0.31	0.36	0.47	0.90	1.23	1.47	61.16	69.41	82.51
	0.30	0.36	0.42	0.53	1.08	1.35	1.78	69.11	75.04	89.00
	0.40	0.41	0.47	0.59	1.23	1.45	2.11	78.25	81.46	101.64
	0.50	0.46	0.51	0.62	1.36	1.59	2.33	84.12	89.98	107.85
	0.60	0.50	0.53	0.64	1.48	1.66	2.54	91.59	95.38	110.28
	0.70	0.51	0.52	0.63	1.54	1.66	2.72	91.43	94.86	106.16
	0.80	0.50	0.51	0.61	1.66	1.74	2.80	91.26	94.32	102.18
	0.90	0.50	0.49	0.60	1.73	1.78	3.00	89.63	90.52	96.75
	Avg	0.42	0.46	0.56	1.30	1.51	2.20	78.41	83.95	95.69
NRMSE [-]	0.10	0.15	0.21	0.22	0.04	0.07	0.06	0.43	0.56	0.56
	0.20	0.21	0.24	0.31	0.05	0.07	0.08	0.53	0.61	0.72
	0.30	0.24	0.28	0.35	0.06	0.08	0.10	0.60	0.66	0.78
	0.40	0.27	0.31	0.39	0.07	0.08	0.12	0.68	0.71	0.88
	0.50	0.31	0.34	0.41	0.08	0.09	0.13	0.73	0.79	0.94
	0.60	0.33	0.35	0.42	0.09	0.10	0.14	0.80	0.83	0.96
	0.70	0.34	0.35	0.42	0.09	0.10	0.15	0.80	0.83	0.92
	0.80	0.33	0.34	0.41	0.10	0.10	0.16	0.80	0.82	0.89
	0.90	0.33	0.33	0.40	0.10	0.10	0.17	0.78	0.79	0.84
	Avg	0.28	0.31	0.37	0.07	0.09	0.12	0.68	0.73	0.83
		LIN	QUAD	CUB	LIN	QUAD	CUB	LIN	QUAD	CUB
RMSE	0.10	0.58	0.52	0.39	1.31	1.10	0.94	94.89	79.97	72.00
	0.20	0.67	0.64	0.49	1.48	1.28	1.13	106.67	92.86	85.32
	0.30	0.73	0.73	0.57	1.59	1.45	1.29	114.41	102.74	96.76
	0.40	0.78	0.82	0.66	1.70	1.59	1.49	118.96	113.61	113.68
	0.50	0.80	0.91	0.74	1.77	1.76	1.69	121.94	122.14	125.89
	0.60	0.79	0.95	0.82	1.82	1.89	1.89	121.25	124.12	148.58
	0.70	0.77	0.95	0.82	1.86	2.01	2.03	120.32	122.30	138.57
	0.80	0.75	0.92	0.70	1.88	2.21	2.31	117.48	122.40	120.25
	0.90	0.74	0.86	0.77	1.91	3.15	3.24	112.40	126.33	125.01
	Avg	0.73	0.81	0.66	1.70	1.83	1.78	114.26	111.83	114.01
NRMSE [-]	0.10	0.39	0.34	0.27	0.07	0.06	0.05	0.83	0.69	0.63
	0.20	0.45	0.42	0.33	0.08	0.07	0.06	0.93	0.81	0.74
	0.30	0.49	0.48	0.38	0.09	0.08	0.07	1.00	0.89	0.84
	0.40	0.52	0.55	0.44	0.09	0.09	0.08	1.03	0.99	0.99
	0.50	0.53	0.60	0.49	0.10	0.10	0.09	1.06	1.06	1.10
	0.60	0.53	0.63	0.54	0.10	0.11	0.11	1.05	1.08	1.29
	0.70	0.51	0.63	0.54	0.10	0.11	0.11	1.05	1.07	1.21
	0.80	0.51	0.61	0.46	0.11	0.12	0.13	1.02	1.07	1.05
	0.90	0.49	0.57	0.51	0.11	0.18	0.18	0.98	1.10	1.09
	Avg	0.49	0.54	0.44	0.09	0.10	0.10	0.99	0.97	0.99

the effects of the missing values inserting with the best model (i.e. convolutional autoencoder) are presented with a corruption rate ranging from 0 (i.e. original data set) to 0.9. In average, for each variable, the descriptive statistics remained almost unchanged. The worst effect on the final statistics of the evaluation set was registered on the CO_2 concentration data, hence confirming the findings in Table 4.

Fig. 10 shows exemplary indoor environment data reconstruction over one random day from the evaluation sets. All presented data were corrupted with a masking noise of 0.5. The presented data confirmed once again the results presented in Table 4, namely, that the proposed denoising autoencoder architectures were suitable for filling the missing indoor environment data sequences.

4.4. Data forecasting performance evaluation

As presented in earlier sections, the proposed autoencoder neural networks were implemented to reconstruct short-term indoor environment data gaps, since building data sets often contain missing values that could hinder further energy analysis. Nonetheless, the same models could be also used for short-term indoor environment data forecasting.

Table 6 summarizes the performance of the implemented autoencoders for different predictive horizons. In summary, all the models performed similarly well. However, there was a clear improvement, with respect to the data reconstruction case, in the LSTM configuration. In particular, the RMSE of the LSTM model was 14 % lower for indoor air temperature and 20 % lower for CO_2 concentration. In terms of NRMSE, even in this case, the proposed autoencoder neural networks could forecast relative humidity data better than other variables. In particular, the NRMSE of the LSTM configuration in case of RH data was 59 % lower than T and 80 % lower than CO_2 data (Table 6).

Fig. 11 shows the density distribution of the forecasting residuals produced by the best models identified by the previous table (i.e. LSTM autoencoder), for a predictive horizon of 12 h. Here, the forecasting residuals were computed as the differences between the observed and predicted values on the evaluation set. For each variable, it was possible to note a normal distribution with a mean of zero, hence confirming that the autoencoders were well developed [40]. The presence of extreme values in these pictures could be traced back to undetected outliers, as explained in the previous subsection.

Fig. 12 shows exemplary indoor environment data forecasting over one random day from the evaluation sets. All presented data were corrupted with a masking noise of 0.5 at the end of each time step (predictive horizon of 12 h). The presented data confirmed once again the results presented in Table 6, namely, that the proposed denoising autoencoder architectures were also suitable for the short-term indoor environment data forecasting.

5. Discussion and future work

This study presented a method to reconstruct sub-daily indoor environment data time-series, since short-term missing data are often present in building data sets and they could hinder further energy analysis. Considering that, building energy models usually require inputs at hourly resolution [8] and that, provided a sufficient large training

the observed and inserted corrupted values on the evaluation set. For each variable, it was possible to note a normal distribution with a mean of zero, hence confirming that the autoencoders were well developed [40]. The presence of extreme values in these pictures could be traced back to the undetected outliers, as explained at the beginning of this subsection.

In order to support the previous results, some descriptive statistics of the complete and filled evaluation set are shown in Table 5. Here,

Table 5

Descriptive statistics for the evaluation set before (CR = 0.00) and after (CR > 0.00) missing data reconstruction with the convolutional autoencoder. "Std" and "Avg" stand for standard deviation and average.

CR [-]	T [°C]				RH [%]				CO_2 [ppm]			
	Min	Max	Mean	Std	Min	Max	Mean	Std	Min	Max	Mean	Std
0.00	12.90	31.70	22.68	1.12	0.90	81.30	39.08	10.98	192.50	2,000.00	510.78	122.81
0.10	12.90	31.70	22.67	1.11	0.90	81.30	39.06	10.96	195.00	2,179.84	510.90	121.83
0.20	12.90	31.70	22.67	1.11	0.90	81.30	39.06	10.94	195.00	2,044.92	509.18	120.04
0.30	12.90	30.10	22.67	1.11	0.90	78.45	39.02	11.02	195.00	2,062.05	507.51	116.02
0.40	12.90	30.10	22.70	1.09	0.90	74.80	39.04	11.04	192.50	2,000.00	509.30	116.51
0.50	13.10	30.10	22.68	1.05	0.90	81.30	39.12	10.96	195.00	2,140.17	505.35	107.22
0.60	13.10	30.10	22.68	1.05	0.90	72.20	39.11	10.87	192.50	2,042.48	498.78	99.12
0.70	13.10	28.50	22.70	1.04	0.90	72.20	39.04	10.74	195.00	2,000.00	512.10	105.56
0.80	13.20	28.50	22.69	1.00	0.90	72.20	39.23	10.63	195.00	2,000.00	504.87	96.45
0.90	13.20	28.40	22.65	0.99	0.90	72.20	39.13	10.73	195.00	1,376.00	508.78	91.04
Avg	13.03	29.91	22.68	1.06	0.90	76.22	39.09	10.88	194.44	1,982.83	507.42	108.20

Table 6

Performance of denoising autoencoder neural networks for forecasting indoor environment data. "CONV", "FEED" and "LSTM" stand for convolutional, feed-forward and LSTM denoising autoencoder. "Avg" stands for average. PH is the predictive horizon.

	PH [h]	T [°C]			RH [%]			CO_2 [ppm]		
		CONV	FEED	LSTM	CONV	FEED	LSTM	CONV	FEED	LSTM
RMSE	2.50	0.19	0.18	0.17	0.89	0.97	0.89	25.62	23.86	25.34
	5.00	0.31	0.30	0.29	1.47	1.44	1.37	43.00	42.38	39.33
	7.00	0.42	0.41	0.40	1.72	1.83	1.76	55.62	55.29	56.11
	9.50	0.50	0.48	0.46	2.15	2.21	2.18	72.37	72.84	74.21
	12.00	0.55	0.52	0.52	2.47	2.45	2.41	81.27	82.55	82.11
	14.50	0.62	0.60	0.59	2.88	2.85	2.67	102.71	102.40	104.51
	17.00	0.73	0.66	0.66	2.84	2.88	2.81	108.13	107.22	107.79
	19.00	0.74	0.64	0.63	2.93	2.95	2.84	102.40	101.58	102.02
	21.50	0.75	0.60	0.60	3.01	3.01	3.18	96.89	96.14	96.88
	Avg	0.53	0.49	0.48	2.26	2.29	2.23	76.45	76.03	76.48
NRMSE	2.50	0.13	0.12	0.11	0.05	0.05	0.05	0.22	0.21	0.22
	5.00	0.20	0.20	0.19	0.08	0.08	0.08	0.37	0.37	0.34
	7.00	0.28	0.27	0.27	0.10	0.10	0.10	0.48	0.48	0.49
	9.50	0.33	0.32	0.31	0.12	0.12	0.12	0.63	0.63	0.65
	12.00	0.36	0.35	0.35	0.14	0.14	0.14	0.71	0.72	0.71
	14.50	0.41	0.40	0.40	0.16	0.16	0.15	0.89	0.89	0.91
	17.00	0.48	0.44	0.44	0.16	0.17	0.16	0.94	0.93	0.94
	19.00	0.49	0.42	0.42	0.17	0.17	0.16	0.89	0.88	0.89
	21.50	0.50	0.40	0.40	0.17	0.17	0.18	0.84	0.84	0.84
	Avg	0.35	0.32	0.32	0.13	0.13	0.13	0.66	0.66	0.67

set of data, there are no significant performance drops by modeling occupant behavior data with between minute-wise and hourly time resolutions [68], it was opted for a 30 min frequency to conduct further models' development. An important contribution of this paper is the analysis of autoencoder neural networks for the reconstruction of different types of single indoor environment streams, measured over multiple years. Accordingly, this fills an important research gap in the related literature, since existing studies on missing data inserting either focused on reconstructing a single type of data stream, they required a number of specific information from additional sensors (e.g. weather data) or they were limited by the size of the available data set.

As presented in Section 3, data were split in three sets before normalization. Model training was performed using a training set, the optimal model configuration was chosen based on performance on the validation set and the final data reconstruction accuracy was tested using the evaluation set. Approximately 94,085 full days of observations were available, which makes this study – as far as the authors know – the largest of its kind for indoor environment data time-series reconstruction purposes. In order to guarantee a significant generalization of the models, it was decided to take an extensive evaluation set. Accordingly, approximately 2.7 million data points were used for models' evaluation. Improved performance of the final models could be achieved by introducing the dimension of each data set as an additional hyperparameter. However, this choice would have led to additional computational costs and, therefore, it was not pursued.

In total, 1,890 hyperparameter combinations were explored by applying a grid search. For experiments run in CPU mode on a single laptop with Intel(R) Core(TM) i7-8850H CPU (2.60 GHz) processor, the time required for training each model was approximately 0.98, 0.12 and 5.62 s/epoch, respectively for the convolutional, feed-forward and LSTM autoencoders. Additionally, the time required for reconstructing the first eight days of the evaluation set (with the pre-trained models) was approximately 0.066, 0.036 and 0.264 s, respectively. In particular, approximately 7,000 core hours of GPU and CPU computations were exploited using compute sources granted at RWTH Aachen University. However, since the size of the training set was set to 30 % of the total data set, the time required for simulation would have decreased in case less data had been available. Due to the limited computational resources, the comprehensive model development that requires extensive computations could not be performed within the building control. Rather, the building control should include the model evaluation which consists of computationally efficient single forward pass, that required around 0.065, 0.036 and 0.256 s for a single day reconstruction, respectively for the convolutional, feed-forward and LSTM autoencoders. The model development, which includes model training and validation, should be conducted as a single-time offline procedure and the final model should be coupled with room control, as already proposed by Romana Markovic [7] and Bauer et al. [69].

This work provided important insights regarding the occurrence of the neural networks saturation for analytics related to building's performance. The consequence of model saturation was that the weights were
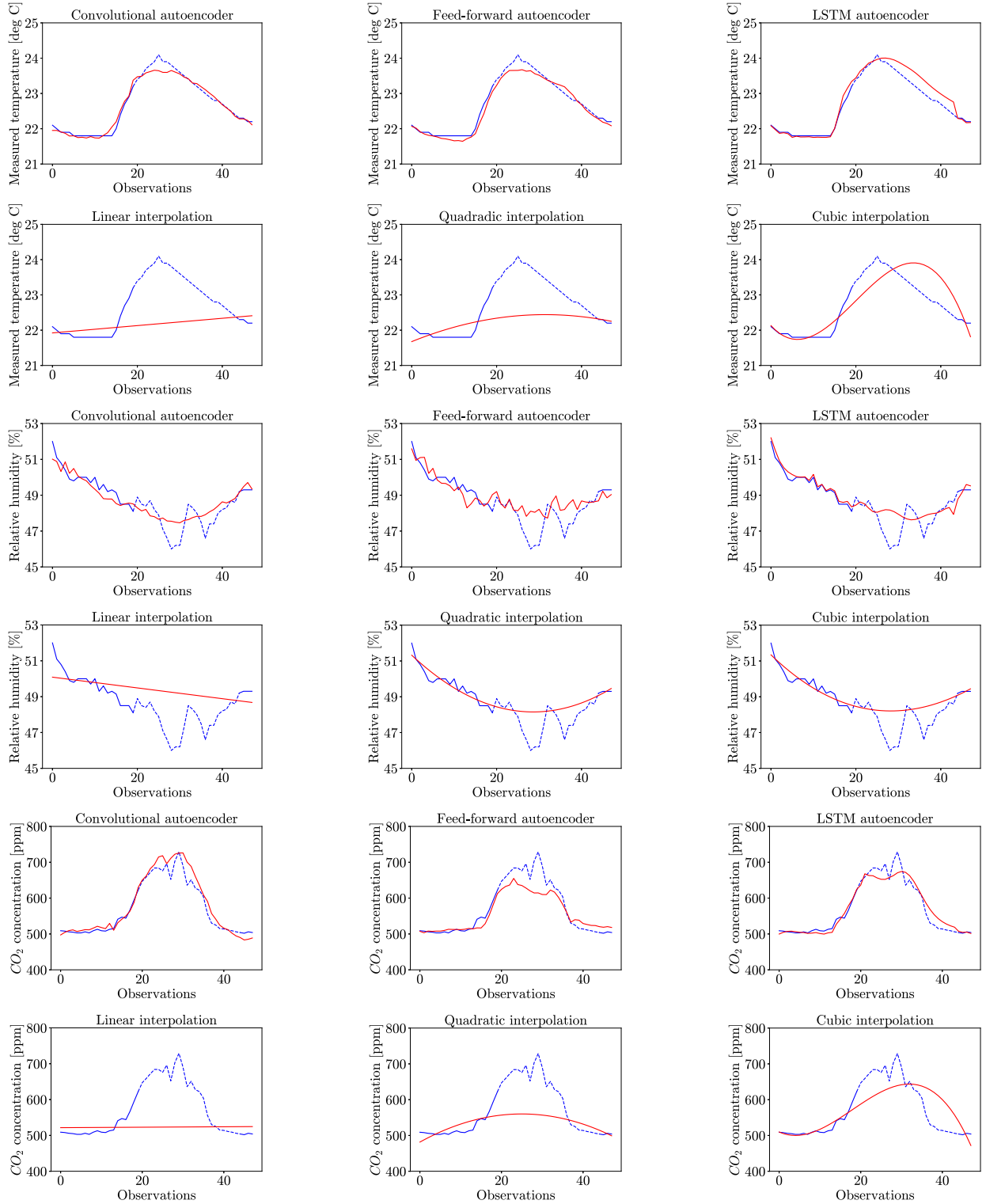
Fig. 10. One day-long indoor environment data reconstruction. Blue colored line represents the real data. Hashed blue colored line represents the missing data. Red colored line represents the reconstruction of the whole day with the adopted model. Observations were sampled to 30 min steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

not updated due to the vanishing gradient problem (Section 4.2), which led always to identical predicted values. This problem was observed in case of the convolutional and feed-forward autoencoders, while it was not detected in case of the LSTM autoencoders. The suitable approach to tackle the model saturation was explored in the existing literature on computer vision and general machine learning. Even though all

the computed saturation metrics were well above the defined saturation limit (0.1), the SAT decreased with the corruption rate for both indoor air temperature and CO_2 concentration and remained almost unchanged for relative humidity. This was inconsistent with the increasing RMSE trend of the latter variable (Table 4). It can be concluded that the above saturation metric could not be used as an additional
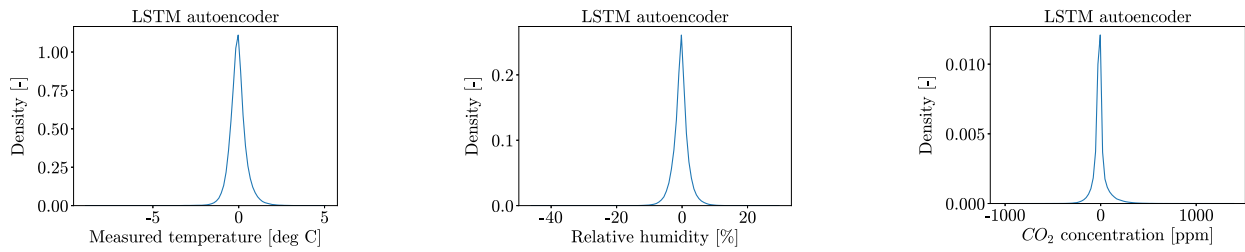
Fig. 11. Density distribution of the forecasting residuals for a corruption rate of 0.5.
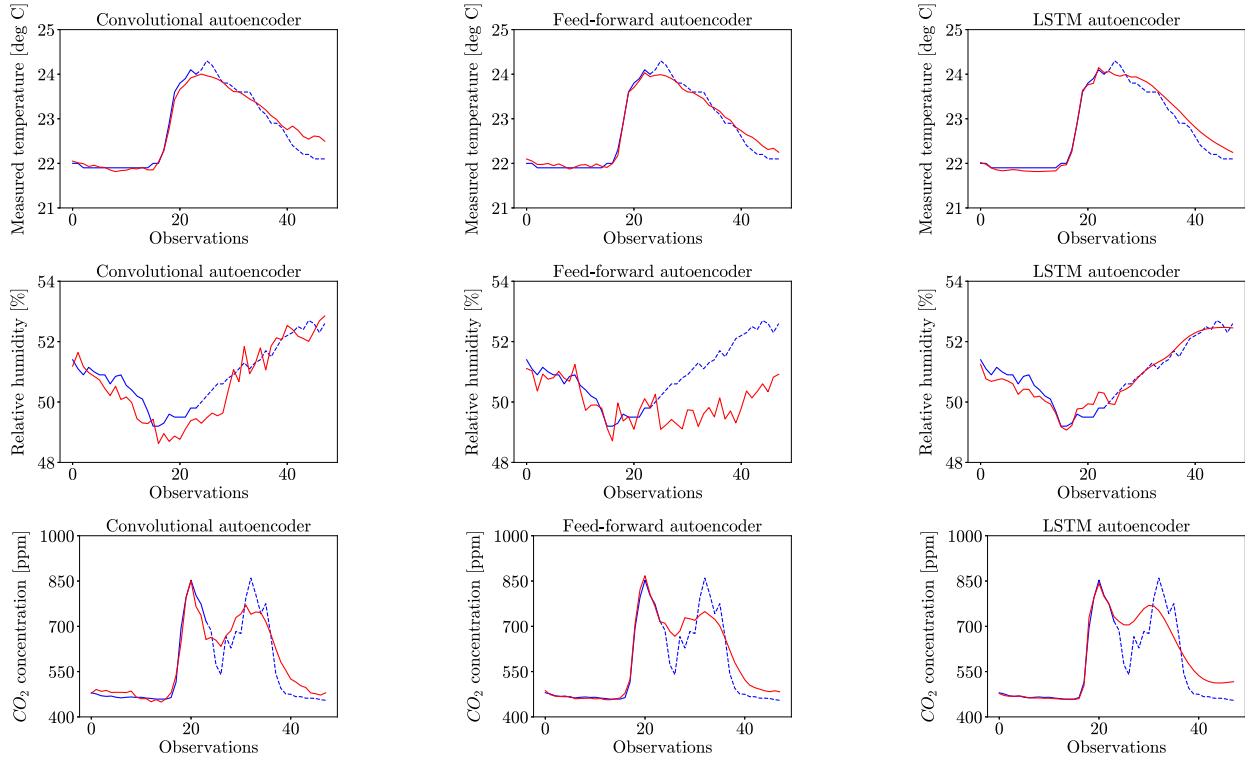

Fig. 12. One day-long indoor environment data forecasting. Blue colored line represents the real data. Hashed blue colored line represents the missing data. Red colored line represents the reconstruction of the whole day with the adopted model. Observations were sampled to 30 min steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

performance measure, since it was dependent also on the sequence-variability of the original data. In particular, the worst saturation performance on CO_2 data could be explained with the presence of more extreme values and frequent peaks. This could explain the reason why LSTM neural networks suffered saturation issues in the paper proposed by Markovic et al. [36]. In this earlier study, an LSTM-based model was applied to plug-in loads data and saturation occurred in more than 70 % of the trained configurations. This could be caused by the larger data imbalance of the plug-in loads and the extreme values. Namely, similarly to plug-in loads, the time-series of CO_2 concentration consist of frequent peaks and extreme values, which showed to be a particular complexity to be considered when using the LSTM for building's energy analysis.

The data reconstruction analysis, applied to the non corrupted data, revealed that the proposed autoencoder neural networks, especially the LSTM configuration, could accurately capture the indoor environment patterns. Accordingly, this represents a significant practical potential for the inclusion of these methods in the real time building control. This could be used for anomaly detection purposes, by identifying data sequences with atypical behavior (e.g. noisy data, sensors' malfunctioning) (Fig. 8). On the other hand, the performance of the convolutional configuration spiked out when a certain masking noise was applied to

the evaluation sequences. It can be, therefore, stated that the spatial correlations of input data were more important than the temporal ones, when a gap-filling method was investigated.

The NRMSE analysis established that relative humidity data patterns were, in general, easier to detect by the proposed models. The lowest performance was observed in case of the CO_2 data, due to the presence of more extreme values and frequent peaks compared to the other variables. However, even in the last case, the proposed ANNs could reconstruct the missing sequences with a significant less average error than then adopted baseline approach (approximately 30 % less in terms of RMSE). Additionally, even the CO_2 sequences could be reconstructed with RMSE below 80 *ppm*, which indicates that the reconstruction performance could even be in similar accuracy range as the measuring tolerance of some commercial CO_2 sensor devices.

Similarly to other papers that analyzed the problem of missing data inserting for building monitoring data [11,15,40], ignoring outliers, the reconstruction residuals along the inserted values lay within a normal distribution with a mean of zero. The same model architecture could reproduce a similar distribution for each of the analyzed indoor environment stream, which represents one of the novelty of this work. Additionally, the overall descriptive statistics of the evaluation set were not affected by simulating the missing values with the proposed

method. However, the standard deviation of CO_2 concentration data was slightly underestimated for high corruption rates (CR > 0.5).

The focus on reconstructing short missing sequences was motivated by several reasons. Firstly, more than 60 % of the missing sequences from building control ranged between one and six hours [38,39]. Motivated by this common yet rarely addressed problem in room automation, we aimed to explore computational approaches that could tackle this issue. Secondly, one of the starting hypotheses was, that the sub-daily time-series could be the longest sequences from room control that pose stationary time-series properties. As a consequence, the meaningful research boundaries were set and the computational methods to tackle this particular problem were investigated. Nonetheless, the future research activities should include as well the analysis of the potential and goodness of multi-day time-series reconstruction and forecasting.

A further analysis on the models revealed that the proposed method could forecast the indoor air temperature data even better than calibrated Modelica-based building performance simulation tools applied in other studies [68]. The temporal correlations of input data gained significant importance with respect to the reconstruction case, placing the LSTM configuration on a slightly better performance level than the convolutional one. In this regard, a denoising autoencoder which relies, at the same time, on LSTM and convolutional units could further increase the predictive accuracy of the model.

One of the possible limitations of this study is a direct consequence of the training process. The proposed autoencoders were, indeed, implemented to capture information related to the daily trends of the observed variables. Accordingly, day-ahead data sequences cannot be reconstructed with the current training scheme. Future work should, therefore, evaluate the implemented autoencoders for reconstructing and forecasting energy and environmental data time-series over longer time horizons and analyze their performance on different periods. Another limitation of this work is due to the used data set. Despite the extension of the evaluation set, it is not yet clear how the models would behave when applied to monitoring data from other domains. However, it may perhaps be observed, that the quality of the same data set was not affected by climatic and environmental factors. In case of data collected in Germany in and between the years 2014 and 2017, no extreme weather conditions, floods or hurricanes were indeed registered. Having that in mind, the model applicability for the unusual or extreme weather conditions, such as electricity outage, extreme heat or cold was not investigated[1]. In order to increase the generalization capability of the developed methods, the inclusion of monitoring data collected in multiple buildings with significant differences in thermal mass and design should be further researched. In this regard, as a future extension of this work, the proposed pre-trained algorithms might be used to improve generalization to other domains, in combination with a suitable domain adaptation procedure. Finally, in order to facilitate the real-life applications of the models, future studies should focus on the optimization of a single autoencoder for multiple time-series.

6. Conclusion

The aim of this paper was to develop an approach for reconstructing short-term indoor environment data time-series. For that purpose, three autoencoder neural networks models were implemented and polynomial interpolation methods were evaluated for baseline comparison. The evaluation of models' performance was conducted using indoor air temperature, relative humidity and CO_2 concentration data. The key findings could be summarized as follows:

- The proposed method outperformed polynomial interpolation models for filling sub-daily environmental data gaps.

- For the indoor air temperature, relative humidity and CO_2 concentration data, the differences between the observed and inserted missing values (i.e. reconstruction residuals) had a normal distribution with a mean of zero.
- The standard deviation of the filled data set might be slightly underestimated if the proposed method is used to reconstruct sub-daily CO_2 concentration data gaps.
- In average, relative humidity data were easier to reconstruct, while CO_2 concentration data were more challenging for the models.
- The developed method could be also used for predicting the indoor environment data with high accuracy, over the multi hour time horizon.
- The implementation of normalized initial weights, ReLU activation function and batch-normalization could avoid saturation issues for ANNs.

In summary, the proposed method has a practical potential for the inclusion in the real-time building control, as back-up option in case of sensor failure. However, further evaluations should be carried out, such as the applicability to other domains or to multiple time-series. Furthermore, the same models might be used to complete indoor environment data sets with small gaps (i.e. < 24 h), improving the performance of later applied data-driven models. In this case, further evaluation about the applicability on longer periods should be carried out.

Declaration of competing interest

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

Acknowledgments

Appendix

A.1. Data cleaning

This Appendix provides additional information about the adopted data cleaning process.

A.1.1. Outliers

Outliers were detected favoring model generalization, rather than accuracy. The aim of this paper was, indeed, to provide a tool to reconstruct indoor environment time-series, independently of the type and quality of data. Ma et al. [16] applied the IQR method for the reconstruction of building electric power data, defining as outlier every value out of the following range:

$$[Q1 - 1.5 \cdot IQR; \ Q3 + 1.5 \cdot IQR], \tag{10}$$

where Q_1 is the first quartile of the data set, Q_3 is the third quartile, IQR is the difference between the third and first quartile. Data out of the previous interval were replaced with the nearest IQR limit [16]. However, since the generalization characteristics of ANNs depend on

---

[1] A relevant work of the identification of such events based on extreme weather and environmental conditions was presented by Meier et al. [70]
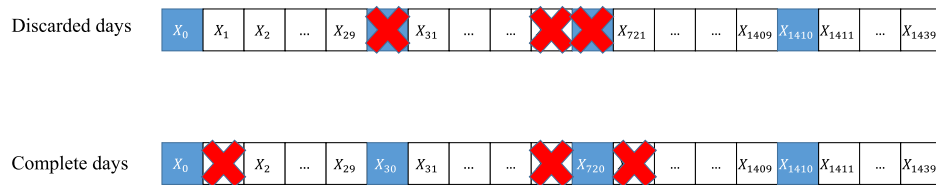
Fig. 13. A visual representation of the discarded and complete days. White blocks are minute-wise observations. Blue blocks are observations with 30-minutes frequency resolution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 7
Descriptive statistics for the data set before (raw data) and after [5,16] outliers detection. Std and L/U IQR stand respectively for standard deviation and lower/upper IQR limit.

	T [°C]			RH [%]			CO_2 [ppm]		
	Raw data	[16]	[5]	Raw data	[16]	[5]	Raw data	[16]	[5]
Min	6	19.4	6	0.5	1.45	0.5	0	265	192
Max	2.34E+16	25.8	37.1	2.53E+03	72.25	99.4	1.31E+04	737	2,000
Mean	6.47E+10	22.64	22.63	3.75E+01	37.52	37.5	5.16E+02	509.12	516.3
Median	22.7	22.7	22.7	36.4	36.4	36.4	491	491	491
Std	3.60E+13	1.22	1.4	1.18E+01	11.01	11	1.25E+02	100.27	124.25
L IQR	19.4	19.4	19.4	1.45	1.45	1.45	265	265	265
U IQR	25.8	25.8	25.8	72.25	72.25	72.3	737	737	737
Outliers	/	2.54E+06	358	/	1.71E+03	313	/	4.18E+06	307

Table 8
Overview of the preprocessed data set.

	T	RH	CO_2
Frequency [min]	30	30	30
Expected days	125,646	125,646	125,646
Discarded days	94,265	94,294	94,294
Complete days	31,381	31,352	31,352

the noise included in the training data [71], the authors decided not to follow this approach at the expenses of an overall reduced accuracy [71]. Accordingly, outliers were detected based on theoretical limits fixed by Markovic et al. [5] in a different study, where a subset of the same data set was analyzed. Therefore, temperature was established between −10 and +40 °C, based on the plausible range for the continental climate in Germany [5]. Relative humidity was set between 0 and 100 % [5], while CO_2 concentration was assumed to be between 0 and 2,500 ppm [5]. Table 7 summarizes descriptive statistics for the data set before and after outliers detection, based on the methods proposed in the literature [5,16]. The IQR method proposed by Ma et al. [16] seemed to oversimplify the problem, by identifying as outliers a wide range of values (Table 7). It was, therefore, opted for the other approach [5].

A.1.2. Missing values

Based on the logging frequency and monitoring duration, it could be expected that around 181 million sets of observations were collected for each variable from 2014 to 2017. Of the latter, only 73 million data points were correctly recorded for T (59.6% error rate), while 70 million for both RH and CO_2 (61.3% error rate). In order to increase the computational efficiency of the models, frequency was reduced from minute-wise to 30 min, leading to approximately 2.3 million data points for each variable. For the missing values handling, a complete case analysis approach was adopted, where only full day of observations with the current resolution were considered. Hence, the number of available monitoring points per variable were reduced to 1.5 million data. Accordingly, from the starting 376,938 daily observations, models were applied only to 94,085 days (75 % error rate). An overview of the missing values handling strategy is presented in Table 8 and in Fig. 13.
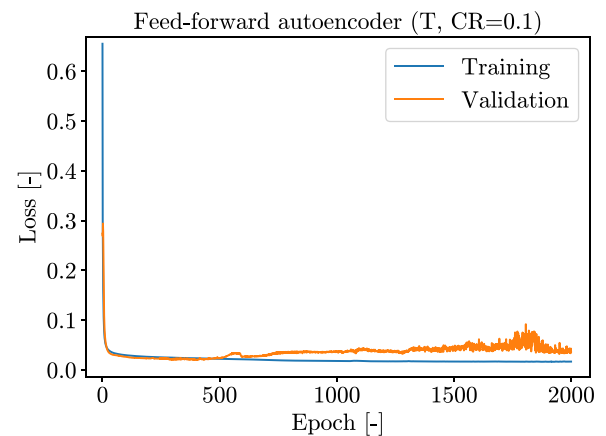

Fig. 14. Example of a training vs validation curve with overfitting.

A.2. Overfitting

Fig. 14 shows the training vs validation curve of the feed-forward model applied to the indoor air temperature data (CR = 0.1), in case of overfitting.

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. European Union 61 (2018) (2018) 75–76, http://data.europa.eu/eli/dir/2018/844/oj.

[2] N. Fumo, A review on the basics of building energy estimation, Renew. Sustain. Energy Rev. 31 (2014) 53–60, https://doi.org/10.1016/j.rser.2013.11.040.

[3] I. Khan, A. Capozzoli, S.P. Corgnati, T. Cerquitelli, Fault detection analysis of building energy consumption using data mining techniques, Energy Procedia 42 (2013) 557–566, https://doi.org/10.1016/j.egypro.2013.11.057.

[4] 2009 ASHRAE Handbook-Fundamentals, Vol. 404, ASHRAE Inc, 2009, pp. 636–8400, http://www.ashrae.org.

[5] R. Markovic, E. Grintal, D. Wölki, J. Frisch, C. van Treeck, Window opening model using deep learning methods, Build. Environ. 145 (2018) 319–329, https://doi.org/10.1016/j.buildenv.2018.09.024.

[6] F. Causone, S. Carlucci, M. Ferrando, A. Marchenko, S. Erba, A data-driven procedure to model occupancy and occupant-related electric load profiles in residential buildings for energy simulation, Energy Build. 202 (2019) 109342, http://dx.doi.org/10.1016/j.enbuild.2019.109342.

[7] R. Markovic, Generic Occupant Behavior Modeling for Commercial Buildings Doctoral thesis, RWTH Aachen University, 2020.

[8] A. Chong, K.P. Lam, W. Xu, O.T. Karaguzel, Y. Mo, Imputation of missing values in building sensor data, in: ASHRAE and IBPSA-USA SimBuild 2016 Building Performance Modeling Conference, Salt Lake City, UT, http://www.ibpsa-usa.org/index.php/ibpusa/article/view/386/372.

[9] J.S. Haberl, T.A. Reddy, D.E. Claridge, W.D. Turner, D.L. O'Neal, W.M. Heffington, Measuring energy-saving retrofits: experiences from the Texas LoanSTAR program, 1996, http://dx.doi.org/10.2172/219427, United States.

[10] A.J. Zapata-Sierra, A. Cama-Pinto, F.G. Montoya, A. Alcayde, F. Manzano-Agugliaro, Wind missing data arrangement using wavelet based techniques for getting maximum likelihood, Energy Convers. Manage. 185 (2019) 552–561, http://dx.doi.org/10.1016/j.enconman.2019.01.109.

[11] L.M. Candanedo, V. Feldheim, D. Deramaix, Reconstruction of the indoor temperature dataset of a house using data driven models for performance evaluation, Build. Environ. 138 (2018) 250–261, http://dx.doi.org/10.1016/j.buildenv.2018.04.035.

[12] A. Rahman, V. Srikumar, A.D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, Appl. Energy 212 (2018) 372–385, http://dx.doi.org/10.1016/j.apenergy.2017.12.051.

[13] J. Loy-Benitez, S. Heo, C. Yoo, Imputing missing indoor air quality data via variational convolutional autoencoders: Implications for ventilation management of subway metro systems, Build. Environ. 182 (2020) 107135, http://dx.doi.org/10.1016/j.buildenv.2020.107135.

[14] Z. Wang, L. Wang, Y. Tan, J. Yuan, Fault detection based on Bayesian network and missing data imputation for building energy systems, Appl. Therm. Eng. 182 (2021) 116051, http://dx.doi.org/10.1016/j.applthermaleng.2020.116051.

[15] J. Peppanen, X. Zhang, S. Grijalva, M.J. Reno, Handling bad or missing smart meter data through advanced data imputation, in: 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference, 2016, pp. 1–5, http://dx.doi.org/10.1109/ISGT.2016.7781213.

[16] J. Ma, J.C.P. Cheng, F. Jiang, W. Chen, M. Wang, C. Zhai, A bi-directional missing data imputation scheme based on LSTM and transfer, Energy Build. 216 (2020) 109941, http://dx.doi.org/10.1016/j.enbuild.2020.109941.

[17] L. Zhang, A pattern-recognition-based ensemble data imputation framework for sensors from building energy systems, Sensors 20 (2020) 5947, http://dx.doi.org/10.3390/s20205947.

[18] M. lbeigi, M. Ghomeishi, A. Dehghanbanadaki, Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm, Sustainable Cities Soc. 61 (2020) 102325, http://dx.doi.org/10.1016/j.scs.2020.102325.

[19] G. Zhang, C. Tian, C. Li, J.J. Zhang, W. Zuo, Accurate forecasting of building energy consumption via a novel ensembled deep learning method considering the cyclic feature, Energy 201 (2020) 117531, http://dx.doi.org/10.1016/j.energy.2020.117531.

[20] X.J. Luo, L.O. Oyedele, A.O. Ajayi, O.O. Akinade, H.A. Owolabi, A. Ahmed, Feature extraction and genetic algorithm enhanced adaptive deep neural network for energy consumption prediction in buildings, Renew. Sustain. Energy Rev. 131 (2020) 109980, http://dx.doi.org/10.1016/j.rser.2020.109980.

[21] N. Somu, G.R. M R, K. Ramamritham, A hybrid model for building energy consumption forecasting using long short term memory networks, Appl. Energy 261 (2020) 114131, http://dx.doi.org/10.1016/j.apenergy.2019.114131.

[22] Y. Gao, Y. Ruan, C. Fang, S. Yin, Deep learning and transfer learning models of energy consumption forecasting for buildings with poor information data, Energy Build. 223 (2020) 110156, http://dx.doi.org/10.1016/j.enbuild.2020.110156.

[23] J.R.S. Iruela, L.G.B. Ruiz, M.C. Pegalajar, M.I. Capel, A parallel solution with GPU technology to predict energy consumption in spatially distributed buildings using evolutionary optimization and artificial neural networks, Energy Convers. Manage. 207 (2020) 112535, http://dx.doi.org/10.1016/j.enconman.2020.112535.

[24] D.H. Tran, D.L. Luong, J.S. Chou, Nature-inspired metaheuristic ensemble model for forecasting energy consumption in residential buildings, Energy 191 (2020) 116552, http://dx.doi.org/10.1016/j.energy.2019.116552.

[25] D.K. Bui, T.N. Nguyen, T.D. Ngo, H. Nguyen-Xuan, An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings, Energy 190 (2020) 116370, http://dx.doi.org/10.1016/j.energy.2019.116370.

[26] Y. Tian, J. Yu, A. Zhao, Predictive model of energy consumption for office building by using improved GWO-BP, Energy Rep. 6 (2020) 620–627, http://dx.doi.org/10.1016/j.egyr.2020.03.003.

[27] F. Qian, W. Gao, Y. Yang, D. Yu, Potential analysis of the transfer learning model in short and medium-term forecasting of building HVAC energy consumption, Energy 193 (2020) 116724, http://dx.doi.org/10.1016/j.energy.2019.11672.

[28] C. Zhou, Z. Fang, X. Xu, X. Zhang, Y. Ding, X. Jiang, Y. ji, Using long short-term memory networks to predict energy consumption of air-conditioning systems, Sustainable Cities Soc. 55 (2020) 102000, http://dx.doi.org/10.1016/j.scs.2019.102000.

[29] Y. Meng, T. Li, G. Liu, S. Xu, T. Ji, Real-time dynamic estimation of occupancy load and an air-conditioning predictive control method based on image information fusion, Build. Environ. 173 (2020) 106741, http://dx.doi.org/10.1016/j.buildenv.2020.106741.

[30] K. Yan, A. Chong, Y. Mo, Generative adversarial network for fault detection diagnosis of chillers, Build. Environ. 172 (2020) 106698, http://dx.doi.org/10.1016/j.buildenv.2020.106698.

[31] K. Yan, J. Huang, W. Shen, Z. Ji, Unsupervised learning for fault detection and diagnosis of air handling units, Energy Build. 210 (2020) 109689, http://dx.doi.org/10.1016/j.enbuild.2019.109689.

[32] C. Hengda, C. Huanxin, L. Zhengfei, C. Xiangdong, Ensemble 1-D CNN diagnosis model for VRF system refrigerant charge faults under heating condition, Energy Build. 224 (2020) 110256, http://dx.doi.org/10.1016/j.enbuild.2020.110256.

[33] Z. Zhou, G. Li, J. Wang, H. Chen, H. Zhong, Z. Cao, A comparison study of basic data-driven fault diagnosis methods for variable refrigerant flow system, Energy Build. 224 (2020) 110232, http://dx.doi.org/10.1016/j.enbuild.2020.110232.

[34] M. Han, R. May, X. Zhang, X. Wang, S. Pan, D. Yan, Y. Jin, A novel reinforcement learning method for improving occupant comfort via window opening and closing, Sustainable Cities Soc. 61 (2020) 102247, http://dx.doi.org/10.1016/j.scs.2020.102247.

[35] A. Das, M.K. Annaqeeb, E. Azar, V. Novakovic, M.B. Kjærgaard, Occupant-centric miscellaneous electric loads prediction in buildings using state-of-the-art deep learning methods, Appl. Energy 269 (2020) 115135, http://dx.doi.org/10.1016/j.apenergy.2020.115135.

[36] R. Markovic, M.K. Annaqeeb, J. Frisch, C. Treeck, E. Azar, Day-ahead prediction of plug-in loads using a long short-term memory neural network, Energy Build. 234 (2021) 110667, http://dx.doi.org/10.1016/j.enbuild.2020.110667.

[37] R. Markovic, J. Frisch, C. Treeck, Learning short-term past as predictor of window opening-related human behavior in commercial buildings, Energy Build. 185 (2019) 1–11, http://dx.doi.org/10.1016/j.enbuild.2018.12.012.

[38] D.E. Claridge, C. Hui, Missing data estimation for 1–6 h gaps in energy use and weather data using different statistical methods, Int. J. Energy Res. 30 (2006) 1075–1091, http://dx.doi.org/10.1002/er.1207.

[39] J.C. Baltazar, D.E. Claridge, Restoration of short periods of missing energy use and weather data using cubic spline and Fourier series approaches: qualitative analysis, 2002, Energy Systems Laboratory, Texas A & M University, http://hdl.handle.net/1969.1/4575.

[40] C. Fan, F. Xiao, Y. Zhao, J. Wang, Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data, Appl. Energy 211 (2018) 1123–1135, http://dx.doi.org/10.1016/j.apenergy.2017.12.005.

[41] Y. Liu, Z. Pang, M. Karlsson, S. Gong, Anomaly detection based on machine learning in IoT-based vertical plant wall for indoor climate control, Build. Environ. 183 (2020) 107212, http://dx.doi.org/10.1016/j.buildenv.2020.107212.

[42] D.B. Araya, K. Grolinger, H.F. ElYamany, M.A.M. Capretz, G. Bitsuamlak, Collective contextual anomaly detection framework for smart buildings, in: 2016 International Joint Conference on Neural Networks, IJCNN, 2016, pp. 511–518, http://dx.doi.org/10.1109/IJCNN.2016.7727242.

[43] A. Legrand, B. Niepceron, A. Cournier, H. Trannois, Study of autoencoder neural networks for anomaly detection in connected buildings, in: 2018 IEEE Global Conference on Internet of Things, 2018, pp. 1–5, http://dx.doi.org/10.1109/GCIoT.2018.8620158.

[44] J. Loy-Benitez, Q. Li. K. Nam, C. Yoo, Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation, Sustainable Cities Soc. 52 (2020) 101847, http://dx.doi.org/10.1016/j.scs.2019.101847.

[45] I. Goodfellow, Y. Bengio, A Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[46] B.K. Beaulieu-Jones, J.H. Moore, Missing data imputation in the electronic health record using deeply learned autoencoders, Pac. Symp. Biocomput. 22 (2017) 207–218, http://dx.doi.org/10.1142/9789813207813_0021.

[47] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (12) (2010) http://portal.acm.org/citation.cfm?id=1953039.

[48] J.P. Fütterer, A. Constantin, Energy concept for the E.ON ERC main building, 2014, E.ON Energy Research Center 4, https://publications.rwth-aachen.de/record/443118.

[49] J.P. Fütterer, A. Constantin, M. Schmidt, R. Streblow, D. Müller, E. Kosmatopoulos, A multifunctional demonstration bench for advanced control research in buildings—Monitoring, control, and interface system, in: IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, 2013, pp. 5696–5701, http://dx.doi.org/10.1109/IECON.2013.6700068.

[50] G. Bode, J. Fütterer, D. Müller, Ein demonstrator für innovative Technologien und Konzepte, in: TGA-Kongress 2016 Berlin, 2016, http://publications.rwth-aachen.de/record/660706/files/Vortragsfolien.pdf.

[51] M. Baranski, A. Kümpel, T.P. Schild, M.H. Schraven, F. Stinner, T.P.B. Storek, D. Müller, J. Fütterer, Ein flexibles lebendes Labor für die Entwicklung und Erprobung von Cloud-basierten Regelungsalgorithmen für die Gebäudeautomation, in: Deutscher KäLte- und Klimatechnischer Verein, DKV e.V.: Tagung 2018, 2018-11-21-2018-11-23, Aachen, 2018, http://publications.rwth-aachen.de/record/750585.

[52] The HDF Group, Hierarchical Data Format, version 5, 1997-2020. http://www.hdfgroup.org/HDF5/.

[53] Python Core Team, Pickle — Python object serialization, 2015, Python Software Foundation. https://docs.python.org/3/library/pickle.html.

[54] K. Sang, K. Jong, Statistical data preparation: Management of missing values and outliers, Korean J. Anesthesiology 70 (4) (2017) 407–411, http://dx.doi.org/10.4097/kjae.2017.70.4.407.

[55] M. Rätz, A.P. Javadi, M. Baranski, K. Finkbeiner, D. Müller, Automated data-driven modeling of building energy systems via machine learning algorithms, Energy Build. 202 (2019) 109384, http://dx.doi.org/10.1016/j.enbuild.2019.109384.

[56] T. Jayalakshmi, A. Santhakumaran, Statistical normalization and back propagation for classification, Int. J. Comput. Theory Eng. 3 (1) (2011) 1793–8201, http://www.ijcte.org/papers/288-L052.pdf.

[57] M. Ferrando, A. Marchenko, S. Erba, F. Causone, S. Carlucci, Pattern recognition and classification for electrical energy use in residential buildings, in: Building Simulation 2019, Rome, 2019, http://dx.doi.org/10.26868/25222708.2019.210750.

[58] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous distributed systems, 2016, CoRR, https://arxiv.org/abs/1603.04467v2.

[59] F. Chollet, Keras, 2015, https://keras.io.

[60] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, CoRR (2015) https://arxiv.org/abs/1502.03167v3.

[61] L. Prechelt, Early stopping-but when? in: Neural Networks: Tricks of the Trade, Springer, Berlin, Heidelberg, 1998, pp. 55–69.

[62] I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning, in: Proceedings of the 30th International Conference on Machine Learning, Vol. 28, (3), 2013, pp. 1139–1147, http://proceedings.mlr.press/v28/sutskever13.html.

[63] K. Diederik, J. Ba, Adam: A method for stochastic optimization, 2014, https://arxiv.org/abs/1412.6980v9.

[64] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305, http://dx.doi.org/10.5555/2188385.2188395.

[65] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of Machine Learning Research, Vol. 9, 2010, pp. 249–256, http://proceedings.mlr.press/v9/glorot10a.html.

[66] R. Grosse, Introduction to Neural Networks and Machine Learning, in: CSC 321 Winter 2018, http://www.cs.toronto.edu/rgrosse/courses/csc321_2018.

[67] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proceedings of the 30th International Conference on Machine Learning, Vol. 28, 2013, http://robotics.stanford.edu/amaas/papers/relu_hybrid_icml2013_final.pdf.

[68] R. Markovic, E. Grintal, A. Nouri, J. Frisch, C. Treeck, Right on time - exploring suitable time discretization for occupant behavior co- simulation, in: Building Simulation 2019, Rome, 2019, http://dx.doi.org/10.26868/25222708.2019.210166.

[69] P. Bauer, M. Aleksy, C. Gross, T. Gamer, Sicherung der praktischen Anwendbarkeit der Algorithmen zur Nutzerverhaltensmodellierung durch Intergration in Moderne GA-Systeme, in: Bauphysiktage Weimar 2019, 2019.

[70] A. Meier, T. Ueno, M. Pritoni, Using data from connected thermostats to track large power outages in the United States, Appl. Energy 256 (2019) 113940, http://dx.doi.org/10.1016/j.apenergy.2019.113940.

[71] D.K. Chaturvedi, Soft Computing: Techniques and Its Applications in Electrical Engineering, Vol. 103, Springer, 2008, pp. 51–85, http://dx.doi.org/10.1007/978-3-540-77481-5.