Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model

Zhen Fang, Nicolas Crimier, Lisa Scanu, Alphanie Midelet, Amr Alyafi, Benoit Delinchant

Highlights

Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model

Zhen Fang,Nicolas Crimier,Lisa Scanu,Alphanie Midelet,Amr Alyafi,Benoit Delinchant

- End-to-end methodology for multi-zone indoor temperature prediction
- LSTM-based seq2seq model
- Cross-series learning strategy
- Tailor-made metric adapted to the special characteristic of indoor temperature
- Evaluation of the forecasting capacity of the model with regard to the forecast horizon
- Benchmark analysis with Prophet model and a seasonal naive model
- Comprehensive evaluation of the model performance

Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model⋆

Zhen Fang[a,*,1], Nicolas Crimier[a], Lisa Scanu[a], Alphanie Midelet[a], Amr Alyafi[b] and Benoit Delinchant[b,**]

[a]Probayes, 53 Avenue Jean Kuntzmann, 38330 Montbonnot-Saint-Martin, France
[b]Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab, 38000 Grenoble, France

ARTICLE INFO

Keywords:
Indoor temperature forecasting
Smart building
Energy saving
HVAC
Recurrent neural network
LSTM
Seq2seq model
Multi-step forecasting
Prediction interval (PI)

ABSTRACT

Accurate indoor temperature forecasting can facilitate energy savings of the building without compromising the occupant comfort level, by providing more accurate control of the HVAC (heating, ventilating, and air conditioning) system. In order to make the best use of different input variables, a long short-term memory (LSTM) based sequence to sequence (seq2seq) model was proposed to make multi-step ahead forecasting. The out-of-sample forecasting capacity of the model was evaluated with regard to different forecast horizons by various evaluation metrics. A tailor-made metric was proposed to take account of the small daily-variation characteristic of indoor temperature. The model was benchmarked against Prophet and a seasonal naive model, showing that the current model is much more skillful and reliable in very short-term forecasting. A cross-series learning strategy was adopted to enable multi-zone indoor temperature forecasting with a more generalised model. Furthermore, the uncertainty in model parameters was quantified by prediction intervals created by Monte-Carlo dropout (MC-dropout) technique.

1. Introduction

Energy consumption has long been one of the key concerns facing the crisis of global warming. The building sector accounts for approximately 44% of the total energy consumption and emits more than 123 million tonnes of CO2 in France [1]. On the other hand, people spend most of their time inside of buildings, the living conditions of which need to be guaranteed. Therefore, how to optimise the energy consumption without necessarily compromising thermal comfort level is a crucial issue in the domain of energy management. One efficient measure is to optimise the HVAC system by leveraging the indoor temperature forecasting [2].

Traditionally, physical or semi-physical models were deployed in modelling the indoor temperature, but their input parameters are based often on specific building characteristics and occupant behavior, which are not always easy to access. ([2, 3, 4]).

Nowadays more and more buildings are being equipped with sensors, which can potentially carry valuable information on characteristics of building, outdoor weather conditions (e.g. temperature, wind, solar radiation) and occupant behaviour (e.g. temperature set points, CO2 level). Therefore, in the last decades, many studies have resorted to data-driven models to predict the indoor temperature, electricity load or energy consumption. Generally speaking, two types of models have been explored: statistical model and machine learning (ML) model. In the early 2000s, most of the data-driven models were statistical models, including ARIMA (Auto-regressive Integrated moving average) [5] and exponential smoothing such as Holt-Winters [6]. Traditional ML methods have also been widely tested, the popular ones among which include SVR (support vector regression) [7, 8], random forest [9], XGBoost etc [10]. In the past decade, Neural Network (NN) has gained popularity in the energy domain due to its ease of incorporating exogenous variables, low requirements on feature engineering and capabilities of dealing with interactions among nonlinear features [11, 12, 13]. In the early 2010s, most of the NN models were based on multilayer perceptron (MLP) [8, 14, 15]

⋆This work has been partially supported by MIAI@Grenoble Alpes, France (ANR-19-P3IA-0003)
*Corresponding author
**Principal corresponding author
zhen.fang@grenoble-inp.org (Z. Fang); nicolas.crimier@probayes.com (N. Crimier); lisa.scanu@probayes.com (L. Scanu); alphanie.midelet@probayes.com (A. Midelet); amr.alyafi@gmail.com (A. Alyafi); benoit.delinchant@G2Elab.grenoble-inp.fr (B. Delinchant)
ORCID(s): 0000-0002-5594-969X (Z. Fang)
[1]Completed as part of internship with Probayes and G2Elab.

and Neural Network Auto-regressive model with eXogenous inputs (NNARX) architecture [3, 16, 17], but in recent years, the LSTM and the attention-based architecture have gained great popularity in time series forecasting thanks to its high accuracy and good scalability [18, 19].

Also, the building energy management becomes more complicated with the increasing complexity of buildings, as the temperature pattern tends to vary with different zones of the building [20]. Traditionally, many individual models were deployed to simulate temperatures of different zones, making it operationally difficult to have a centralised control of the whole building. Unlike traditional statistical models which are only trained to predict one target variable (referred to as "single-series learning" in this paper), the NN is able to train multiple related time series altogether in one lumped model, and simultaneously predict the temperature for multiple locations in a building (referred to as "cross-series learning" in this paper). Additionally, this kind of "cross-series learning" can also improve the performance of the model, as many related time series augment the total volumes of the training set, rendering it more robust in handling outliers. With the arrival of the big data era, this method has attracted strong attention among researchers and the industry. For instance, the winner of M4 forecasting competition [21] has used a global hybrid model to cater for characteristics of individual time series. The recent DeepAR model [22] developed by Amazon and a seq2seq model of Uber [18, 23] both have adopted this type of cross-series learning approach.

This paper proposes a seq2seq LSTM model to forecast the indoor temperature of different zones of the building, with the cross-series learning strategy. The whole preprocessing and training procedure is presented through its application on 3-year real operational data collected from the GreEn-ER building, a dynamic and low-energy consumption building located in the heart of Grenoble city, France.

2. Related studies

Many studies have been performed in leveraging the NN model to predict the indoor temperature or energy consumption. NN model is a special ML model inspired by the network of biological neurons. The model usually consists stacks of layers of artificial neurons (or units) and during a computation many neurons work in parallel to produce the result. Compared to other ML methods, no explicit mathematical model structure is enforced in the first place and the model tends to find the optimal solution via a heuristic approach.That's also why NN model is often called a "black-box" model, which is extremely flexible to approximate non-linear functions. Some of the previous studies are listed below.

Attoue et al. [14] conducted a study in 2018, which leveraged a MLP model to predict the indoor temperature of an old building, using indoor and outdoor temperature, humidity as well as solar radiation data. The study concluded that the MLP model can provide good forecasts up to 4-hour ahead for the reason that the Mean Squred Error (MSE) is below 1. Mba et al. [24] used MLP model as well for indoor temperature forecasting with a time lead between one day and one month. They concluded that the model gave good results up to one-day ahead, with a correlation coefficient of 0.985. Afroz et al. [16] used a NARX model to predict multi-zone indoor temperatures, and claimed that the model gave good performances up to 28-day ahead, with a MSE score of 0.083. However, inspection of their qualitative results reveals that the model fails to predict the peaks, and since the test sample is very small, the model has not been thoroughly tested with more anomalous events when the HVAC system is turned off or functions differently. Mustafaraj et al. [17] used a NNARX model to predict multi-step ahead indoor temperature and relative humidity for an open office, with a recursive strategy. They concluded that the model can provide accurate prediction up to 3-hour ahead, as after that the error exceeds a predefined threshold 0.78. Similarly, Lu and Viljanen [3] also trained a NNARX model to predict indoor temperature and relative humidity in a test house, where one-month of data was collected, with a resolution of 15 minutes. The indoor temperatures were forecasted four-step ahead (up to 1-hour ahead) with a recursive strategy. A fast-adopted delta method was used to generate the prediction interval , the quality of which is visually evaluated based on its coverage rate. A LSTM model was used in the study of Wang et al. [25] to predict 24h-ahead thermal load. They concluded that LSTM is only suitable to predict very short term prediction (one to several hours) , but not 24h-ahead, by comparing the result with XGBoost. Xu et. al [26] also used a LSTM model to forecast very short-term (5min and 30-min-ahead) indoor temperatures in a public building, and claimed that their model outperforms other common machine learning methods, like Support Vector Machine (SVM). Similarly, the study of Muzaffar et al. [27] used a LSTM to predict electricity load for multiple forecast horizons and they concluded that the LSTM outperforms other statistical models (ARMA, SARIMA and ARMAX) for short-term prediction up to 48h ahead. However, the model structure was not flexible enough in dealing with output sequences which have a different sequence length than the input sequence. Besides, only historical variables were used.

The shortcomings of many of the existing studies can be summarised in the following aspects:

- Most of the studies have only used historical records to predict the future indoor temperature, and among the ones using both historical and future input variables, their related preprocessing procedure was often not explicitly explained [26].
- The time series used to train the NN model were often very short, varying between a few days [28] to a few months [14, 29]. This will to a large extent limit the performance of NN, as it is well known that the NN performs better when it's trained on large datasets [30]. Moreover, the metric score obtained from a small test set can hardly generalise the model performance under different circumstances.
- Most of the studies have benchmarked their NNs with other sophisticated machine learning method, like random forecast [28], XGBoost [25] with the metric of Symmetric Mean Absolute Percentage Errors (MAPE) or Root Mean Squared Error (RMSE). However, surprisingly not many studies have performed a benchmark analysis with the naive method [14], which sometimes is hard to be beaten by even the most advanced ML models [31]. Furthermore, since the indoor temperature does not have a strong daily variation, we need to take a more critical look at those "low" forecast errors: an average RMSE of 0.8 degree might already be too big in comparison to the average daily variation (as reflected in our studies see Figure 7).
- Most of the studies of indoor temperature forecasting target a very short predefined forecast horizon (1h - 1day) [3, 26], instead of exploring the saturating point of the model with regard to the forecast horizon.

The current paper aims to cover the research gaps listed above, and provide a systematic working framework for similar future studies. The main contributions can be summarised as follows:

1. A complete end-to-end methodology, including data preprocessing, model construction, calibration and comprehensive result analysis, was demonstrated with 3-year building operational data.
2. Three different LSTM-based seq2seq architectures were explored and compared under different forecasting contexts. The saturating point of the current model was addressed with regard to the forecast horizon.
3. A tailor-made evaluation metric was designed specifically for multi-zone indoor temperatures forecasting.
4. Benchmark analysis was conducted with Prophet model and a seasonal naive model.
5. A 95% prediction interval was generated to take a better account of model uncertainties.

The rest of this paper is organised as follows: Section 3 introduces the concept of LSTM-based seq2seq model after a technical review of the study problem. Section 4 describes the end-to-end methodology with the experiment on GreEn-ER building. The performance of the model is examined comprehensively in Section 5. Section 6 concludes this paper with suggestions for future studies.

3. Model architecture

3.1. Problem statement

Technically speaking, this study is a multi-step time series forecasting problem, which aims to predict the indoor temperature several time steps ahead. For a time series with strong seasonal patterns, like the indoor temperature, typically we can use previous days (input sequence) of temperature to predict the following days (output sequence). The length of the output sequence is predefined as the forecast horizon, while the length of the input sequence is a parameter that needs to be calibrated according to the nature of the data.

It is also a multivariate time series forecasting problem, in the sense that the model does not only use historical records of the target variable to make the forecast, but also exogenous variables. Depending if they can be forecasted or known in advance or not, the input variables can be splitted into two groups: historical variables (e.g. past indoor temperature of the room) and future variables (e.g. forecasted outdoor temperature, hour of day, etc).

RNN is one type of NN that is specially designed for dealing with sequential data. As illustrated in Figure 1, the recurrent neuron receives at each time step not only the input Xt, but also its own output(or hidden state depends on the complexity of the RNN model) from the previous time step. LSTM is a special type of RNN, which adds a cell state in addition to the hidden state to conserve valuable information along the time sequence. The information and gradient flow is also controlled by a special gate system, which can help cope with the vanishing and exploding gradient

problems much better than the normal RNN model. For further details of the RNN and LSTM model, the readers can refer to [32] and the references therein. In this paper, we will focus rather on the architecture of LSTM-based seq2seq model.

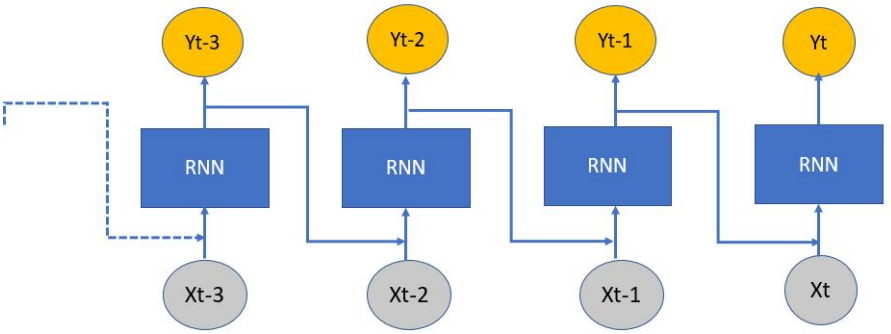


Figure 1: Illustration of RNN architecture: X indicates the input sequence, Y indicates the output sequence, the hidden state is represented by the connections along the time sequence t.

A common practical issue related to the RNN/LSTM model is that it cannot generate the output sequence of different length to the input sequence easily. Many studies have therefore chosen a recursive / iterative strategy, which iteratively uses the prediction at time t+1 as an input variable for the prediction of the next step (t+2). However, this approach tends to accumulate prediction errors along time steps, as well illustrated in the studies of Wen et al.[33]. Another workaround is to directly predict a sequence of future values by using the seq2seq architecture. The seq2seq (or named encoder-decoder) architecture was originally designed for neural machine translation [34], but has emerged as a good architecture in solving various sequential problems, including time series.

The basic idea of this architecture is to use an encoder to extract important features from the input sequence, then pass these features to the decoder to generate the forecast. Although originally the author used LSTM for both the encoder and the decoder component, in practice different types of architectures can be used. For instance, Kim et al. [35] has used a CNN-LSTM architecture to predict residential energy consumption for the next hour with the information collected over the last hour; Wen et al.[33] has proposed a LSTM-MLP architecture for general time series forecasting problems.

3.2. Seq2seq architecture

As mentioned in section 2, many previous studies have already used LSTM model to forecast indoor temperatures, but they were rarely used in a seq2seq architecture. In this study, three basic LSTM-based seq2seq architectures were explored in forecasting the indoor temperatures of the GreEn-ER building. The structures of these architectures are explained in the following sections.

3.2.1. LSTM-Dense model

LSTM-dense model deploys one LSTM as encoder, from which the output of the last time step is then passed to a dense layer (or named fully-connected layer), which functions as a decoder to directly generate a vector of outputs with a length equals to the forecast horizon. The historical variables and future variables are fusioned before being passed into the encoder. The architecture is illustrated in Figure 2.

This architecture is, however, not really suitable for situations when the input sequence is shorter than the output sequence, as then the future variables cannot cover the full forecast horizon. Nevertheless, it is empirically found that the model works better with an input sequence covering at least the full length of the output sequence or a full length of the most important seasonality.

3.2.2. LSTM-LSTM model

Compared to the LSTM-dense model, the LSTM-LSTM model passes the historical and future variables to the encoder and decoder respectively. This structure is more flexible in dealing with cases when the output sequence is of different length than the input sequence. Specifically, the LSTM encoder extracts features from the historical variables,

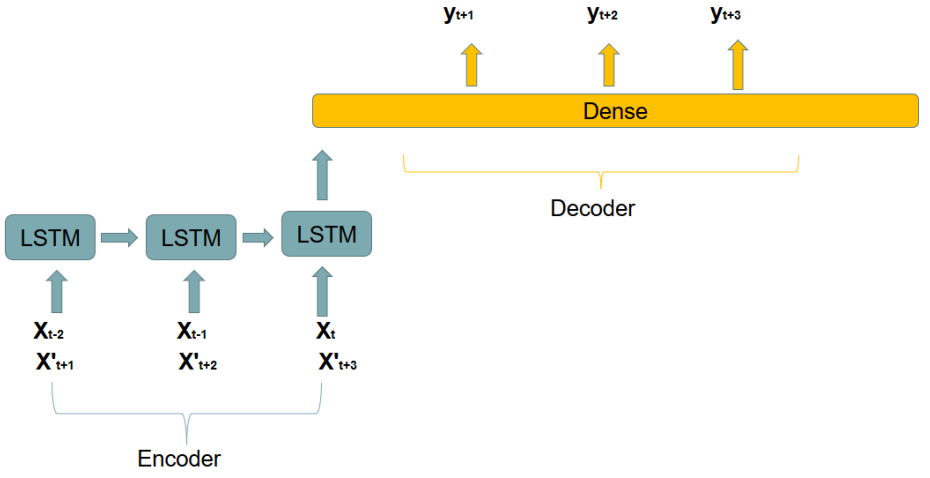


Figure 2: Illustration of LSTM-dense architecture: X indicates the historical variable; X' indicates the future variable, y indicates the model output.

then the hidden state of its last time step is passed to the LSTM decoder. The future variables are passed directly to the decoder at each output time step. A time-wrapped fully-connected layer transforms the output of the LSTM decoder directly to predicted output at each future time step. The architecture is illustrated in Figure 3.

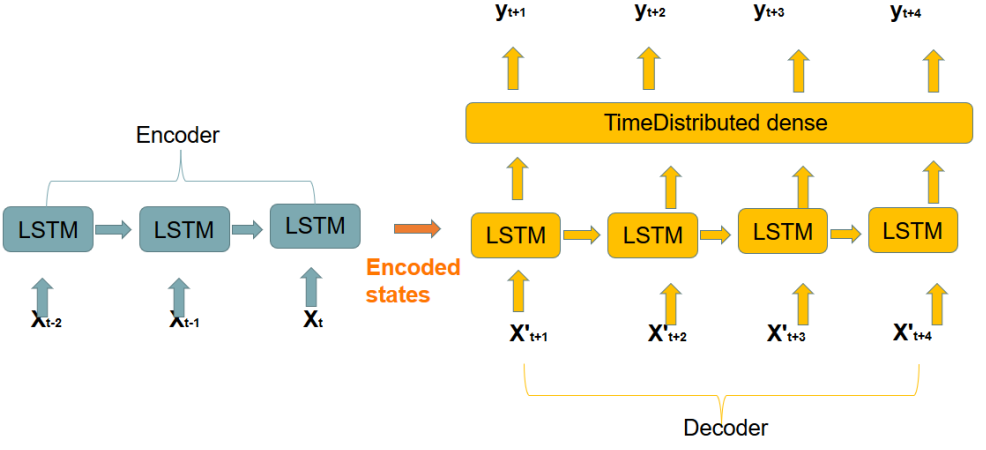


Figure 3: Illustration of LSTM-LSTM architecture: X indicates the historical variable; X' indicates the future variable, y indicates the model output.

3.2.3. LSTM-dense-LSTM model

The LSTM-dense-LSTM architecture can be seen as a variant of the LSTM-LSTM model. The key differences between the two are: a dense layer, with a smaller dimension, is inserted between the encoder and the decoder to further extract information out of the encoded features. Also, the last time step output of the encoder is firstly distilled by a dense layer (with typically a lower dimension) before being concatenated with the future input variables at each future time step. In this way, we ensure that important historical features can be passed successfully to the decoder at each forecast time step. The contraction brought by the dense layer also has a regularization effect against overfitting. The architecture is illustrated in Figure 4.

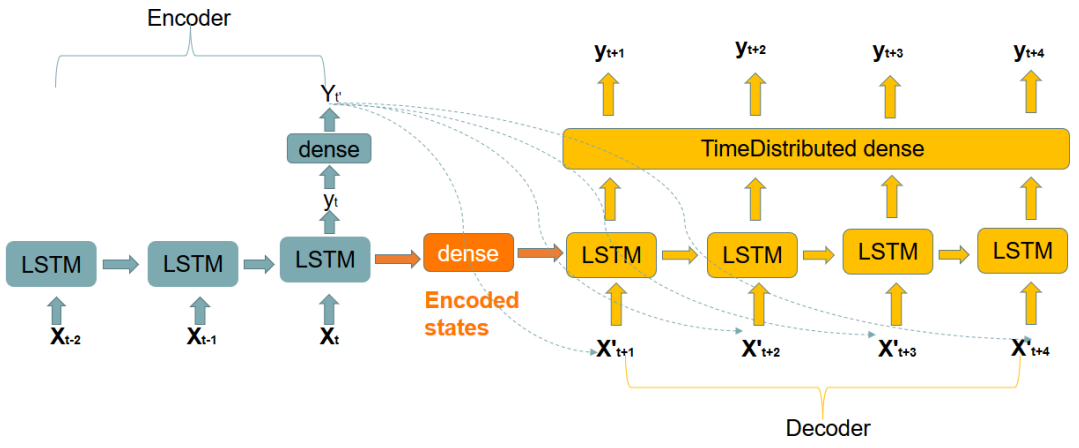


Figure 4: Illustration of LSTM-dense-LSTM architecture: X indicates the historical variable; X' indicates the future variable, y indicates the model output.

4. Case study

4.1. GreEn-ER buildings

GreEn-ER is a 6-floor institutional building, dedicated to foster innovation in energy use and renewable resource research and training. With 4500 m² space per floor, the building receives ca. 2000 persons per day, out of whom about 1500 are students.

Figure 5: GreEn-ER building global overview.

With more than 1000 measuring points in total, all rooms are closely monitored with different sensors. Figure 6 shows the distribution of temperature sensors of the 4th floor. One specific zone called Predis-MHI is especially monitored (lower-left corner of 6). Predis-MHI has an energy control system for each room, based on ventilation and water-to-air heat exchange. Each room is equipped with different sensors, including set points of the room temperature, ventilation flow, ventilation temperature, indoor temperature, CO2 concentration, lighting and occupancy.

4.2. Data description

3 years of data were collected from the GreEn-ER building between 01/01/2017 and 31/12/2019, including:

- 8 rooms of ca. half-hourly indoor temperature
- CO2 values of the room "4A016" with a resolution of 4 minutes
- hourly outdoor temperature

- hourly solar radiation DHI (Diffuse Horizontal Irradiance) and DNI (Direct Normal Irradiance)
- discontinuous room temperature set points of the 8 rooms

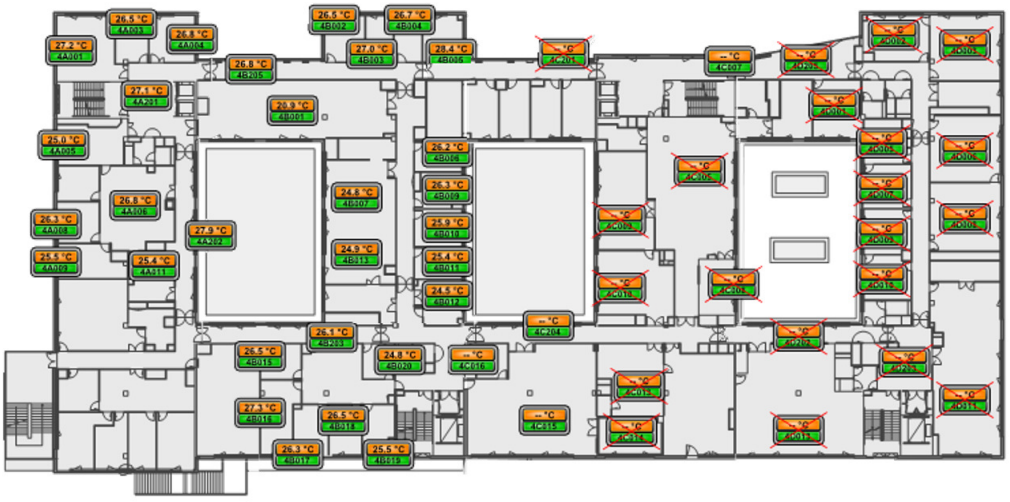


Figure 6: Temperature monitoring on each zone; example of the 4th floor.

In addition to the measured outdoor temperature, a discontinuous, 2-day ahead forecasted time series of outdoor temperature is available as well for the same period. A comparison of the two time series shows that, despite some oscillations in the first half year of 2017, the forecasted time series matches the measured very well, demonstrating that the 2-day ahead weather forecast is of very high accuracy. Although the forecast outdoor temperature should be used as a dominating future variable for the model, the time series is not continuous and of limited horizon, which would limit the testing capacity of the model. Considering that the two time series are of close similarity, the measured outdoor temperature is used in place of the forecasted in this study.

All the data were first scrutinized for outliers, defined as values outside the normal range ($median \pm 3std$). Then they were resampled to an one-hour resolution. The standard deviations of the daily indoor temperatures at different months of the year are illustrated in Figure 7, for the selected 8 rooms.

A scatter plot was used to visualize the correlation relations between the indoor temperature and the other measured exogenous variables (see figure 23). Obviously, the outdoor temperature has the highest positive correlation with the indoor temperature, followed by the solar radiation variables (DNI and DHI).

The CO2 variable does not exhibit a direct relation with the indoor temperature, but as illustrated in figure 24 , it actually has rather a non-linear positive correlation with the indoor temperature at low temperatures. Since CO2 is a good indicator of the occupancy level of the room, the CO2 may have a direct influence on the indoor temperature during the cold season.

4.2.1. Missing values

All input data contain to some degree missing values. An overview of the missing values in the time series of indoor temperature is shown in figure 20. In order to avoid information leaking caused by one-shot interpolation over the whole time series, the missing values were interpolated per sliding window (see chapter 4.3.2) when preparing the input and output sequence vectors. For missing values located in the front or end of the sliding window, a back-filling and forward-filling was implemented respectively.

4.2.2. Seasonality

For classic statistical methods, like ARIMA and exponential smoothing, the time series need to be first deseasonalized. However, this is not always necessary for an NN model if all the series in the dataset follows homogeneous

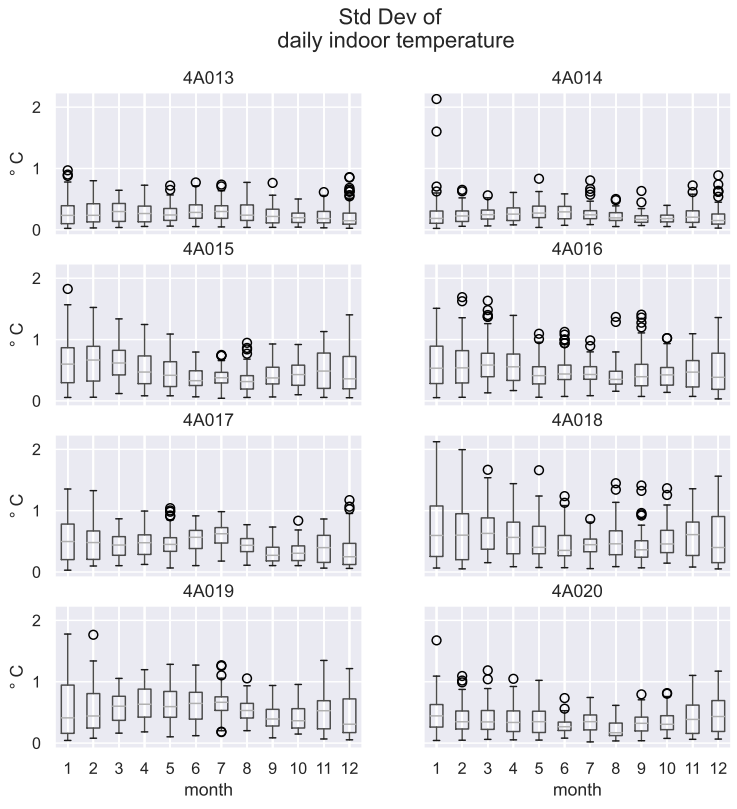


Figure 7: Daily standard deviation of the indoor temperature measured at the 8 rooms.

seasonal patterns, covering the same duration and with sufficient lengths, as demonstrated by the studies of Hewamalage H. et al.[36]. Thus the seasonalities of the indoor temperature in this project were addressed by several selected temporal variables.

Three strong seasonalities are clearly manifested in the time series of indoor temperature: intrayear cycles, intraweek and intraday cycles.

Figure 8 shows that there is a strong difference between the weekdays and the weekends during the summer (July-September) and winter season (December to March), while the difference is much diluted during the mid-season (April-June & October-November). Additionally, no clear difference was found among weekdays, except for the early morning of Monday, during which a transition effect is observed. Hence, a binary indicator was created to encode the weekly seasonality, where the weekend was encoded as 1 and the weekday was encoded as 0.

The hourly and monthly patterns were encoded by a sine-cosine transformation, as illustrated by the formula 1, to ensure a continuous representation of the temporal variable while preserving the distance between adjacent values.

$$x_{sin} = \sin\frac{2\pi x}{T} \quad \text{and} \quad x_{cos} = \cos\frac{2\pi x}{T} \tag{1}$$

T – length of the seasonality feature

x – variable to be transformed

Considering that the HVAC system is turned off during the public holidays, a binary variable was added to indicate the period of public holidays. As the period of Christmas is inherently very different from the other public holidays

due to the long suspension time of the heating, a binary variable was specially designated to the period of Christmas holiday (23/December to 01/January).

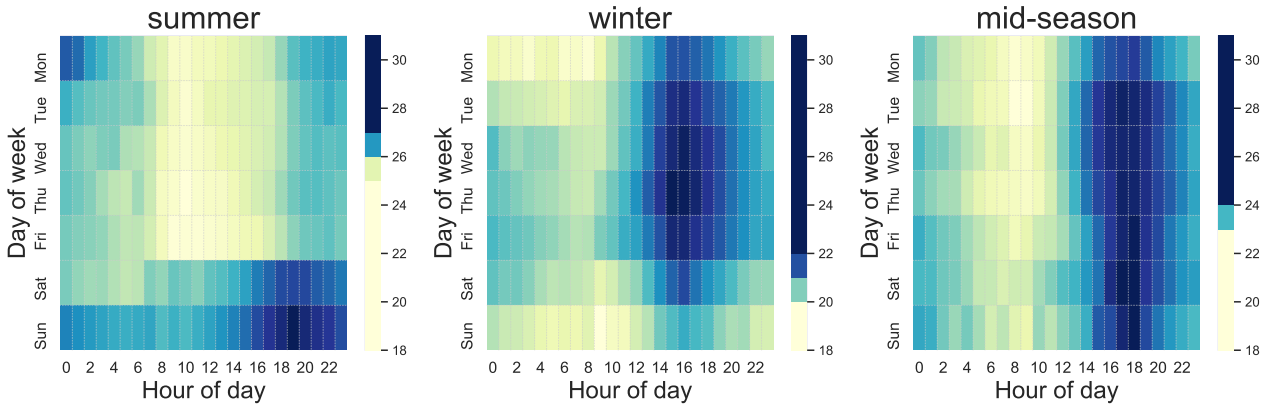


Figure 8: Indoor temperature distribution over week and day, along with the seasons.

4.3. Preprocessing

4.3.1. Normalisation

All the input variables were normalised to the range of [-1, +1] before being fed to the LSTM model. This normalisation method was also tested against another two common methods: normalisation to the range of [0, 1] and standardization ($\frac{x-x_{mean}}{x_{std}}$). The model result later showed that the two normalisation methods gave similar performance, and both were better than the standardization method.

4.3.2. Data preparation

Despite that LSTM is specially designed to treat relatively long sequence data, the model has a capacity limit to deal with extremely long time sequences as a result of numerical instabilities and exploding weights. Therefore, a common practice is to split the very long individual time series into many small time sequences with a sliding window. Besides, it can function as a data-augmentation technique as well. Considering that it is a supervised learning problem, we can prepare two coupled matrices X and Y so that each sample x_i has its correspondent target label y_i. The detailed procedure is explained below:

The historical variables are decomposed into two components: historical indoor temperature and historical exogenous estimators. The whole input feature space is given by $X_i = [x_1, x_2, \ldots, x_t] \in R^{t \times q}$, where t denotes the input sequence length and $q = m + 1$ (m is the number of historical exogenous estimators and 1 represents the historical indoor temperature).

The future variables contain two components as well: temporal variables and future outdoor temperature. Together they form a matrix space of $Q_i = [q_{t+1}, q_{t+2}, \ldots q_{t+k}] \in R^{k \times q}$, where k denotes the forecast horizon and $q = n + 1$ (n denotes the number of temporal variables and 1 represents the future outdoor temperature).

Depending on the specific type of the seq2seq architecture, the historical variables and the future variables are either concatenated to form a total input matrix before being passed directly to the encoder (LSTM-dense), or are passed separately to the encoder and decoder component (LSTM-LSTM, LSTM-dense-LSTM).

Time series forecasting has a strong chronological attribute, which makes it inappropriate to predict old records with recent data. A common practice is to use a sliding window, as illustrated in Figure 9. The input (X) and output(Y) windows slide simultaneously through the whole time series with step size of one hour. In this perspective, historical variables are the ones contained in the time frame of X window, future variables and the target variable are the ones contained in the time frame of the Y window. Since we want to train a global model for all of the 8 rooms, the X and Y matrices of the 8 rooms were shuffled and stacked together to form a lumped X and Y matrix.

As mentioned in section 4.2.1, the missing values were interpolated along with the sliding window to avoid information leaking from the future values.

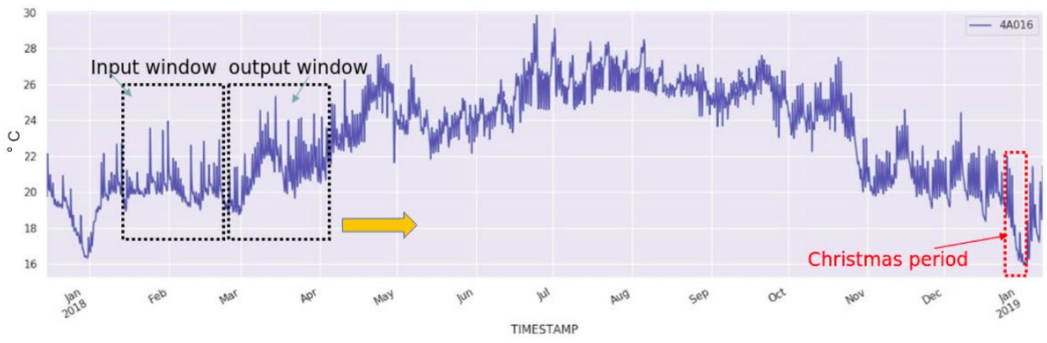


Figure 9: Illustration of the sliding window technique.

4.3.3. Train-dev-test dataset

The 3-year target variables were split into three datasets: the first 2 years were used as the training set, the last year of room '4A016' was used as the test set, and 50-100 (depending on the length of the output sequence) random samples drawn from the last year of the other rooms were used as the validation set. The other exogenous variables were sampled the same way, except that the exogenous datasets were the same for both the validation and the test dataset.

4.4. Model Training

The mean squared error (MSE) loss function was minimised using the Adam optimizer [37], thanks to its excellent performance in handling the complex training dynamics of LSTM and the fast rate of convergence.

The model was trained for two forecast horizons: very short-term (48h-ahead) and short-term (7-day ahead). For 7-day ahead forecasting, it was observed that the exploding gradient problem persists due to the long input sequence. Therefore, the gradient clipping technique was deployed to scale down gradients that exceed the normal range. The maximum gradient was set to 3 based on observations of the gradient norm during training.

The model in this work was developed and evaluated in python language using Pytorch framework.

4.4.1. Tuning of hyperparameters

A grid of hyperparameter values were firstly defined, and a set of random combinations of the hyperparameters were drawn from this space (technically named Random search). For each random set of hyperparameters, the model was evaluated based on its average RMSE on the validation set. Since random search does not try every combination in the set of possible values, it does not guarantee to find the optimal set of hyperparameters. A manual fine-tuning was thus later made for the most sensitive parameters.

The optimal hyperparameter values of the LSTM-dense model are shown below, together with their corresponding search range. The predefined search range covers the most probable searching space of each hyperparameter.

Table 1
Random search of the hyperparameters

hyperparameters	optimal value	search range
initial learning rate for Adam optimizer	0.0025	[0.0005, 0.001, 0.0025, 0.005,0.01]
mini-batch size	128	[32, 64, 128, 256]
dropout probability	0.2	[0, 0.2, 0.4, 0.6, 0.8]
nr. of hidden units	128	[32, 64, 128, 256]
nr. of hidden layer	1	[1, 2]
length of the input sequence	168 (for 7d-ahead forecasting)	[120,168,216,336]

It was discovered that the most important hyperparameter are the learning rate, followed by the number of hidden units, dropout probability, and length of the input sequence.

4.4.2. Overfitting

While tuning the hyperparameters, it was found that the model performed better with a large number of units combined with relatively strong regularization, rather than a small number of units with little regularization. The drawback of using such a large-capacity model was that the model tended to overfit very fast, especially with an efficient optimizer like Adam.

Two different dropout schemes were explored to regularize the model. The first dropout scheme, also the most common one, is to use a different dropout mask at each time step for only the inputs and outputs of the recurrent unit, and no dropout is applied on the recurrent unit [38].

The second dropout scheme is called weight-dropped technique [39], a variant to variational dropout [40], which applies the same dropout mask through the time sequence,as illustrated in Figure 10. The difference between the two is that the variational dropout applies the dropout on the hidden units, while the weight-dropped on the weights within the hidden units directly to further strengthen the regularization.

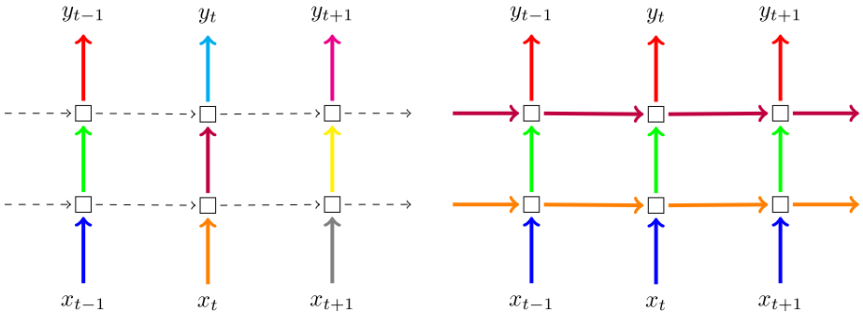


Figure 10: Naive dropout vs. variational dropout in RNN model. source: [40].

The training curve of the 48h-ahead forecasting is shown in Figure 11. The training curve of the 7d-ahead forecasting is shown in appendix A.4. It was observed that the weight-dropped technique can indeed bring stronger regularization on the RNN model, resulting in slightly better performance on the validation set. Compared to the 48h-ahead forecasting, the overfitting was manifested strongly for the 7d-ahead forecasting, indicating that the current model is less capable of forecasting the far future. In addition, the early-stopping mechanism was used as well to further prevent the overfitting.

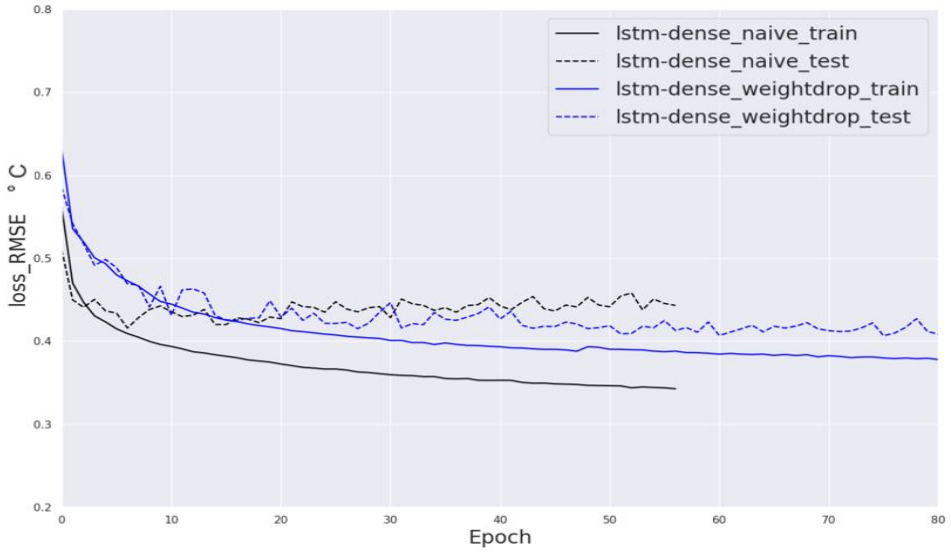
4.4.3. Selection of input variables

As mentioned in Section 4.2, not all collected sensor data were used as input variables for the final model. In this study, the model was constructed with a stepwise approach: the model was firstly constructed with the historical records and the future outdoor temperature (the most important exogenous variable reflected from the correlation analysis A.3). Then the selected temporal variables and other weakly correlated exogenous variables were added one by one according to their contributions to the model performance. In this way, two variables, CO2 and room temperature set points, were excluded from the final model structure, as they both brought a negative contribution to the model.

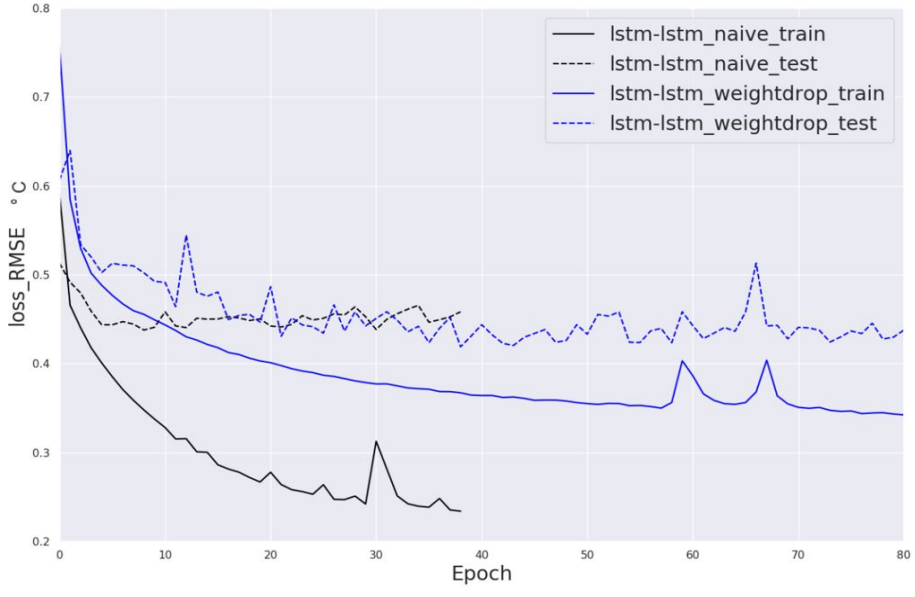
Despite the fact that the CO2 is a strong indicator of the occupancy level of the room, it did not bring extra values in the current model setting. As illustrated in Figure 24, the CO2 has a nonlinear correlation with the indoor temperature during low temperatures and this effect is diminished with the increase of temperature. Since the extremely low temperatures often occur during special events that were already addressed by the holiday and weekend indicators, it is possible that the addition of the CO2 variable was superfluous. Moreover, manifested by its strong intrahour variations, CO2 has a highly volatile nature, indicating that its historical hourly-averaged value would probably bring little information for the upcoming hours.

As illustrated in Figure 22, the set-point values are almost constant throughout the whole winter/summer season, and the increase or decrease of the value does not bring a corresponding change of the indoor temperature. This "strange" behavior comes from the fact that the GreEn-ER building is a self-sufficient building, in which the heating and cooling capacity is to large extent limited by the other factors (e.g. the energy production of the solar panels, the

capacity of the local data centre, etc.) which were not explicitly addressed by the current model.



(a) LSTM-dense



(b) LSTM-LSTM

Figure 11: training and validation loss of two different seq2seq models for 48h-ahead forecasting, with the two dropout schemes.

4.5. Benchmarking

The developed LSTM-based seq2seq model is benchmarked against Prophet and a seasonal naive model.

4.5.1. Naive model

Considering the strong seasonal patterns of the indoor temperature, the naive model was defined as the average temperature of the first two years on the same hour and date.

4.5.2. Prophet model

Prophet model, developed by Facebook research, is known for its ease of use, explicability and flexibility in dealing with seasonal time series. It is therefore used as another benchmark model. The purpose is to test if the LSTM-based seq2seq model is more attractive than an easy-to-use model like Prophet in terms of model performance and generalization capability.

Prophet is an additive time series forecasting model which contains three key components: trend, seasonality and holiday. Three types of seasonalities were modeled explicitly in this project: the daily, weekly and yearly seasonality, among which the weekly seasonality is set as a conditional seasonality of summer, winter and mid-season, as mentioned in section 4.2.2.

Grid search was applied together with a cross-validation procedure to tune the hyperparameters. Since it does not make sense to use recent data to predict historical events, the cross-validation procedure is defined as follows: firstly we define a list of cutoff dates from the last year's time series, then we construct an X and Y vector for each of the cutoff dates. The X vector contains the first two-year's data up to the cutoff date, while the Y vector contains the forecast horizon (7 day) starting from the cutoff date. In this project, the cutoff dates were selected every 7 days from the last year. The model performance was evaluated by the average root mean squared error (RMSE) of all the test sets (Y vectors).

The optimal parameters and their corresponding searching range are listed in Table 2.

Table 2
Calibration of Prophet model

hyperparameters	optimal value	search range
change-point prior scale	0.01	[0.001, 0.01, 0.05, 0.1]
seasonality prior scale	10	[1, 5, 10, 20, 30]
holidays prior scale	20	[5, 10, 20, 30]
weekly seasonality prior	10	[1, 5, 10, 20, 30]
outdoor temperature prior scale(exogenous variable)	30	[5, 10, 30]
mode	multiplicative	[additive, multiplicative]

5. Result analysis

With the trained and well fine-tuned model architectures, we would like to answer the following questions based on the model's performance on the test set:

1. To which forecast horizon can the model give competitive results in comparison to the in-sample seasonal naive model and daily variations of the temperature?
2. Given the same training data, how does the seq2seq model perform in comparison to Prophet model and the seasonal naive model?
3. With a fixed forecast horizon, is there a huge deviation in model performance among the selected seq2seq architectures?
4. How does the cross-series model perform in comparison to the single-series model?

5.1. Performance measures

Four metrics were used to evaluate the model results for each test sample: RMSE, coefficient of determination (R2), mean absolute scaled error (MASE) and mean normalised absolute error (MNAE). The metric-scores over all the samples were averaged to generate the final score. However, the R2 was calculated directly on the total output (flattened matrix) to avoid having too few values during short forecast horizons.

RMSE and R2 are fairly standard evaluation metrics in time series forecasting, so they will not be explained in detail here. MASE, introduced firstly in 2006 by Hyndman et al. [41], is the mean absolute error of the forecast values, divided by the mean absolute error of the in-sample naive forecast. Typically, a value bigger than one will indicate that the current model performs worse than the in-sample seasonal naive model.

As demonstrated in Figure 7, the daily variation of the indoor temperature is fairly small and varies strongly among different rooms. Based on this particular property, a tailor-made metric MNAE (formula 3) was introduced as the mean absolute error normalised by the average daily standard deviation of the target variable.

$$MASE = \frac{\frac{1}{T}\sum_{i=1}^{T}|\hat{y}_i - y_i|}{\frac{1}{P-m}\sum_{t=m+1}^{P}|Y_t - Y_{t-m}|} \tag{2}$$

T – the forecast horizon
y – out-of-sample target value
$\hat{y}$ – out-of-sample prediction
P – time span of the training set
m – time span of the selected seasonality
Y – in-sample target value

$$MNAE = \frac{1}{T}\sum_{i=1}^{T}\frac{|\hat{y}_i - y_i|}{\sigma} \tag{3}$$

T – the forecast horizon
σ – the average daily standard deviation of the room temperature
y – out-of-sample target value
$\hat{y}$ – out-of-sample prediction

5.2. Performance with forecast horizon

The three seq2seq architectures were trained to predict indoor temperatures up to 7 days ahead, while the evaluation metrics were aggregated for the following intermediate horizons: 1h, 6h, 12h, 24h, 48h, 96h, 120h and 168h. The out-of-sample evaluation metrics are plotted versus the forecast horizon in Figure 12, to gauge how the model's performance deteriorates with longer horizons.

In general, all the metrics reflect the same pattern: the performance of the model degrades significantly with time, especially after the first 48 hours. The MASE-score shows that the model produces far better results than the in-sample naive model for all the forecast horizons considered. The MNAE indicates that the MAE of the forecasts is lower than the average daily standard deviation during the first 2-3 days.

Considering that the outdoor temperature is only provided for the first 48 hours during the forecasts, it is important to know if the model performance could be significantly improved or not if a longer sequence of outdoor temperature was available. Thus, a scenario was tested by providing the same model architectures with 7-day ahead outdoor temperature information. The test was only made for the lstm-dense-lstm and lstm-dense architecture, as the lstm-dense-lstm is just a slight variant of the lstm-lstm.

The resultant metrics (Figure 13) show that both models give very similar performance when the outdoor temperature is only provided for the first 48 hours. However, the lstm-dense-lstm architecture exhibits a considerable performance boosting when the 7-day ahead outdoor temperature is provided, while the lstm-dense benefits much less from this valuable information. This difference implies that the seq2seq model can make better use of the future variables when the historical and future variables are fed separately to the encoder and decoder.

The same models were trained again for very short-term forecasting in order to compare the performances among the three seq2seq architectures. The result (Table 3), shows that the lstm-dense model slightly outperforms (8-10%) the other two.

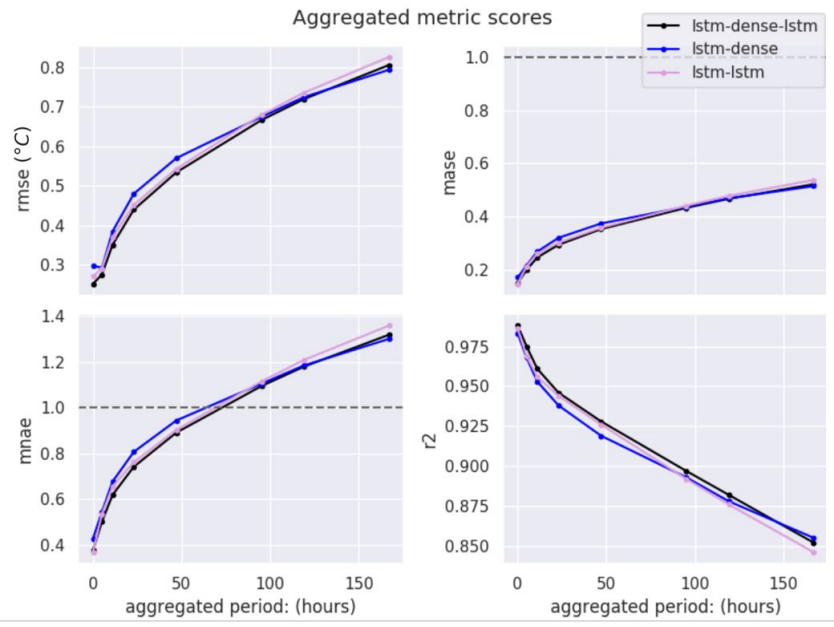


Figure 12: Average aggregated metrics of 7-day ahead indoor temperature forecasting, with weather forecast provided 2-day ahead. The performance threshold in term of MNAE and MASE are indicated by the dash horizontal lines.

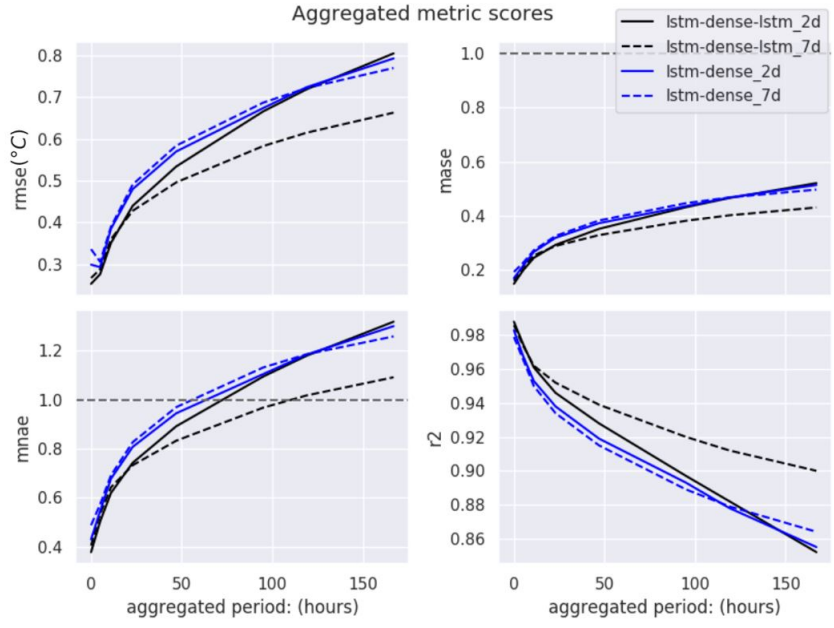


Figure 13: Average aggregated metrics of 7-day ahead indoor temperature forecasting, with weather forecast provided 7-day ahead and 2-day ahead respectively for the lstm-dense-lstm and the lstm-dense model. The performance threshold in term of MNAE and MASE are indicated by the dash horizontal lines.

5.3. Performance comparison with the benchmarks

The lstm-dense-lstm model was then benchmarked against Prophet model and a seasonal naive model mentioned in section 4.5. As Prophet model does not allow cross-series learning, the lstm-dense-lstm model for this test was only trained with one room's (4A016) indoor temperature. Both models were trained with 7-day ahead outdoor temperature.

Prophet model was trained individually up to each cutoff date, and a 7-day ahead forecasting was made for each

Table 3
Comparison of the model performance on very short-term forecasting (48h-ahead)

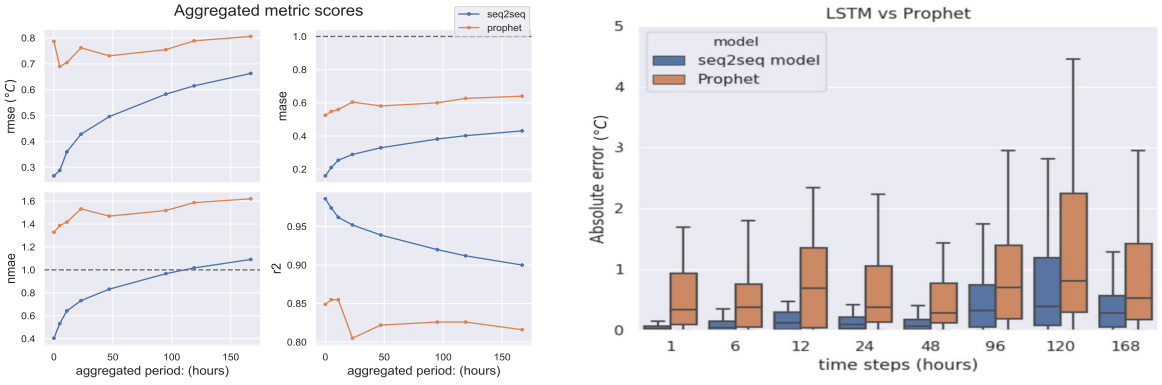
Seq2seq architecture	RMSE-mean (°C)	std dev (°C)	%
LSTM-dense	**0.458**	0.26	ref
LSTM-dense-LSTM	0.493	0.26	7.6%
LSTM-LSTM	0.505	0.28	10.30%

cutoff date. The results of Prophet model were compared with that of the lstm-dense-lstm model for the same forecast periods, both quantitatively (Figure 14) and qualitatively (Figure 15).

Figure 14a shows that the LSTM-based seq2seq model gives a much better performance than Prophet for all the predefined forecast horizons. Although both models beat the in-sample naive model for a large margin, the MNAE-score shows that the average errors of Prophet are all above the daily standard deviation. The boxplot 14b shows that the error variance of Prophet is significantly higher than the seq2seq model, indicating that the model's performance is not very consistent. The benchmarking result is also summarised in Table 4.

Table 4
Tabular illustration of the benchmarking in terms of RMSE (°C)

Forecast horizon (hours)	seq2seq	Prophet	seasonal naive
1h	0.267	0.787	1.275
6h	0.288	0.69	1.013
12h	0.36	0.705	1.013
24h	0.428	0.762	1.012
48h	0.496	0.731	1.011
96h	0.583	0.755	1.011
120h	0.615	0.789	1.011
168h	0.663	0.806	1.011



(a) Evaluation metrics

(b) Boxplot

Figure 14: Quantitative comparison between Prophet model and the seq2seq model trained for room 4A016

5.4. Cross-series learning vs single-series learning

The cross-series model was compared with the single-series model (room 4A016) in predicting the indoor temperatures of all the rooms based on the lstm-dense-lstm architecture. For the single-series model, since no additional test set was available, the validation set was also used for the final test set (this would give the single-series model a better performance on the forecasting of room 4A016 than it should have been).

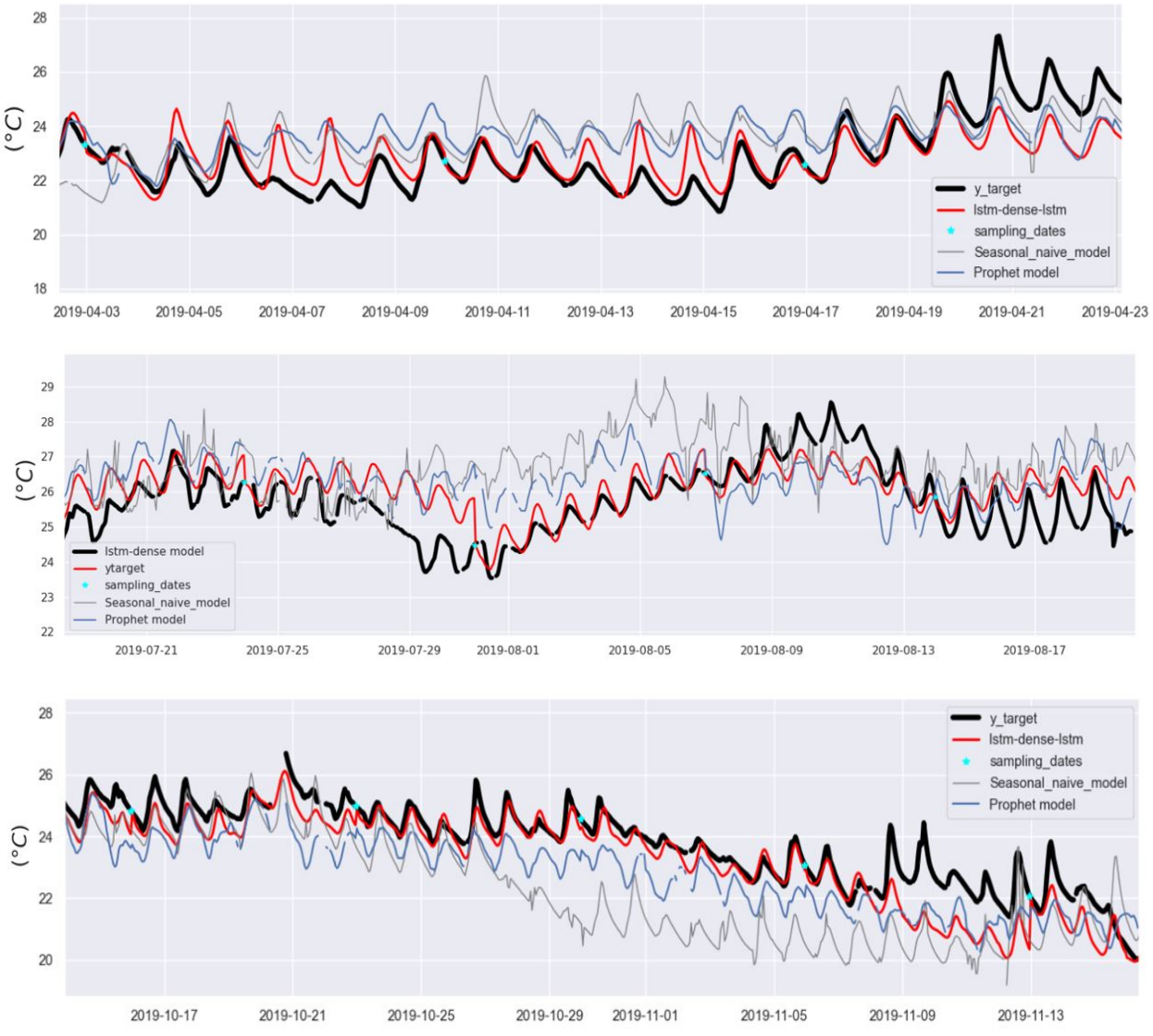


Figure 15: Qualitative comparison among Prophet model (blue line), naive model (grey line) and the seq2seq model (red line) against the ground truth (thick black line). The forecasting was initiated at each sampling date (cutoff date) indicated by the blue points.

The result is illustrated in Figure 16. It shows that the cross-series model outperforms the single-series model in forecasting the temperatures of almost all the rooms, especially during the first 48 hours.The single-room model, despite that it is trained only on one room (4A016), can forecast actually quite well its neighboring rooms (4A018, 4A019) which posses similar time-series pattern. For the same reason, the rooms (4A013, 4A014) which are located far from the training room get much less accurate results from the single-series model. The aggregated RMSE scores of both strategies are also shown in Table 5 for room 4A017.

5.5. Prediction interval (PI)

In order to quantify the level of uncertainty associated with the point forecasts, MC-dropout was applied to generate the prediction interval. According to the studies of Gal et al.[40], the MC-dropout of the NN model can be interpreted as a Bayesian approximation of Gaussian process, which has shown strong generalization ability and scalability. The MC simulation in this study was repeated 100 times during the inference with stochastic variational dropouts, then the

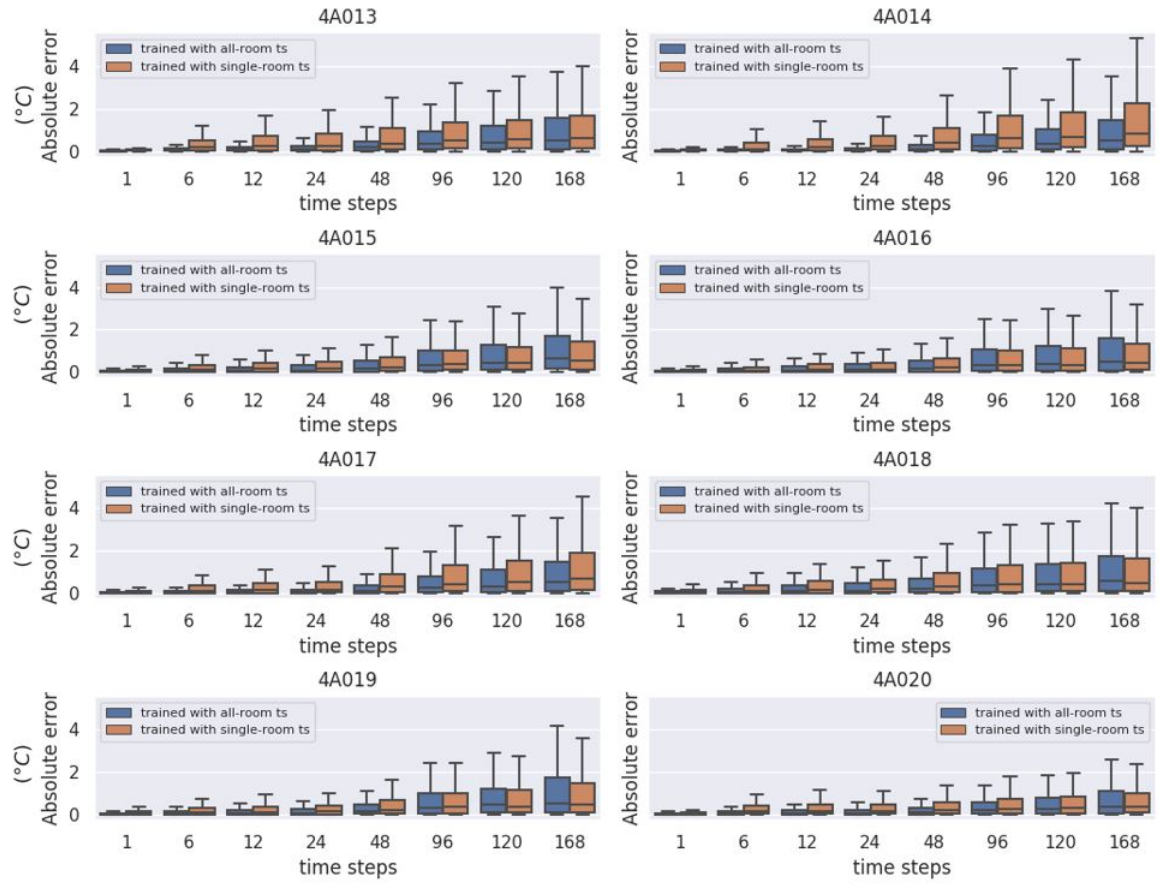


Figure 16: Comparison of the cross-series learning with the single-series learning strategy

Table 5
Average RMSE (°) of the model trained with cross-series learning strategy in comparison to the single-series learning strategy for room 4A017

Forecast horizon (hours)	cross-series	single-series
1h	0.233	0.348
6h	0.232	0.351
12h	0.28	0.438
24h	0.336	0.526
48h	0.407	0.626
96h	0.544	0.75
120h	0.608	0.797
168h	0.718	0.875

model's uncertainty was approximated by the sample variance. Specifically, the 95% prediction interval was calculated as $y_{median} \pm 1.96 * stddev$. A qualitative overview of the prediction interval is shown in section A.5 for three periods of the year (January, July and December) that are most difficult to be forecasted.

The quality of the prediction interval was also evaluated quantitatively by Prediction Interval Coverage Probability

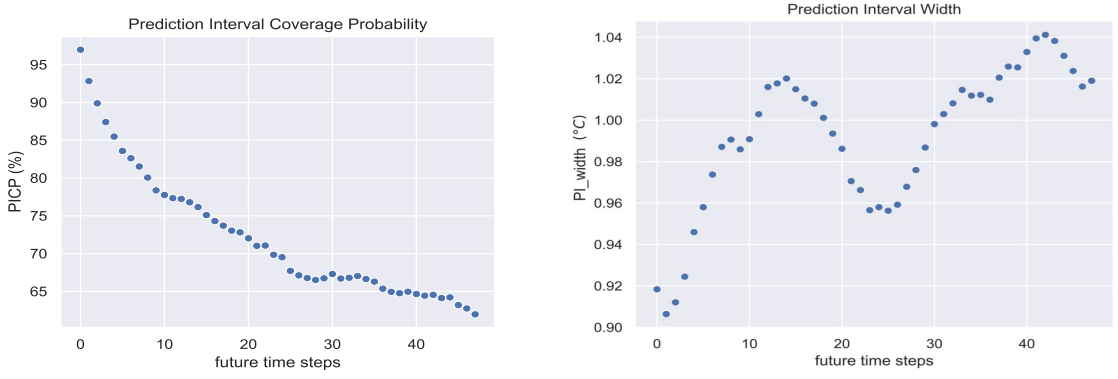
(PICP) and Mean Prediction Interval Width (MPIW) proposed by Khosravi et al. [42].

$$PICP = \frac{1}{T}\sum_{i=1}^{T} C_i \tag{4}$$

c_i – binary value: 1 if the target value y_i is within the PI; otherwise 0

$$MPIW = \frac{1}{T}\sum_{i=1}^{T}(U_i - L_i) \tag{5}$$

U_i – upper limit
L_i – lower limit



(a) PICP (b) MPIW

Figure 17: Quantitative evaluation of the PI

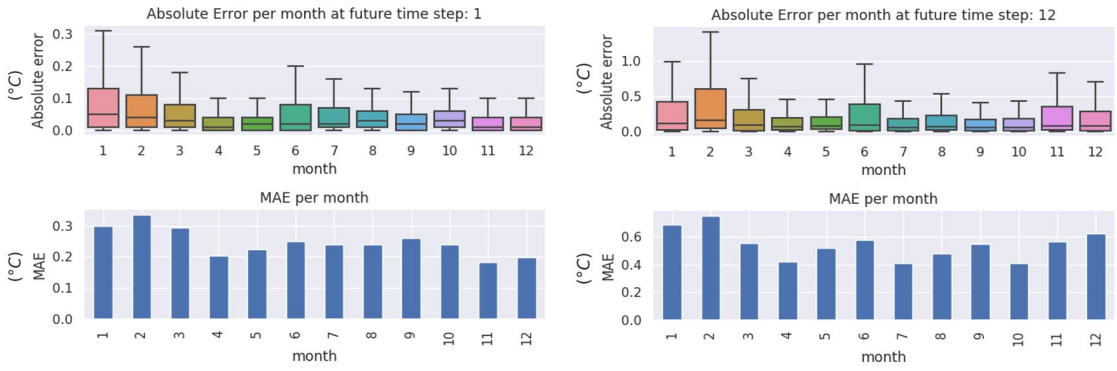
We observe that the average width of the PI increases only slightly with the forecast horizon, while the coverage rate decreases quickly. This implies that the confidence interval cannot fully address the increased uncertainty with time.

The qualitative plot shows that the model can well represent the large uncertainties during certain periods of the year (like the Christmas, the summer period), where the big variance is already manifested in the training dataset. But it is out of the current model's capacity to address the uncertainties related to model misspecifications[43], such as the omission of input variables that are not explicitly modelled, e.g. occupancy level and capacity of the heating and cooling system.

5.6. Residual analysis and model limitations

As the test set covers a full-year time span, we can have a more detailed look at the model performance across the year. A boxplot of the absolute error at one-hour ahead and 12-hour ahead is shown in Figure 18. It can be seen that the model has more difficulties in forecasting during the winter (November - March) period when the heating is on, and the transition period between spring and summer (June) when the cooling system begins to dominate the temperature control of the whole building. Since no input variables have been assigned to address these mechanisms in this project, their impact cannot be well simulated by the current model.

Further, a qualitative check also showed that some large residuals are actually caused by unexpected indoor activities: sudden opening of windows, unexpected energy consumption during the weekend and holidays etc.



(a) one-hour ahead

(b) 12-hour ahead

Figure 18: Distribution of forecast errors over the year

6. Conclusion and suggestion

This paper has presented an end-to-end methodology in predicting multi-zone indoor temperatures with LSTM-based seq2seq model architecture. A tailor-made metric was proposed to take account of the small daily-variation characteristic of indoor temperature. The result shows that provided a 2-day ahead weather forecast information, the model is able to provide skillful forecasting up to 2-3 days ahead. The current model outperforms Prophet and the seasonal naive model by a great margin, especially on the very short-term (48h-ahead) forecast horizon. Also, a cross-series learning strategy was adopted to enable multi-zone indoor temperature forecasting. This greatly enhanced the model performance on different thermal zones. Among the tested three seq2seq model architectures, the LSTM-dense model is more skillful for very short-term forecasting, while the LSTM-dense-LSTM and LSTM-LSTM model can make better use of the future variable, and thus performs better for forecasting with longer horizons.

Furthermore, the uncertainty in model parameters was quantified by prediction intervals created by Monte-Carlo dropout (MC-dropout) technique. The quality of the prediction interval was guaranteed by its high coverage rate and restricted coverage width for very short-term forecasting.

The model performance can be further improved from two perspectives: firstly and most importantly, relevant information could be collected to address important mechanisms that were not covered by the current model: the heating and cooling capacity of the building, the scheduled indoor activities (e.g., big gatherings like meetings, parties, extra loads of the data centre etc). Secondly, modelling is a continuous process. The current model should be kept updated with more advanced model architecture and newly acquired data set. In this study, only LSTM-based seq2seq model was tested. It would be interesting to test other seq2seq architecture, like CNN-LSTM and transformer-based seq2Seq architecture. Future studies will also be devoted to the deployment and integration of the model in the real building energy management system to achieve better energy savings.

A. Appendix

A.1. Indoor temperatures of the modelled 8 rooms

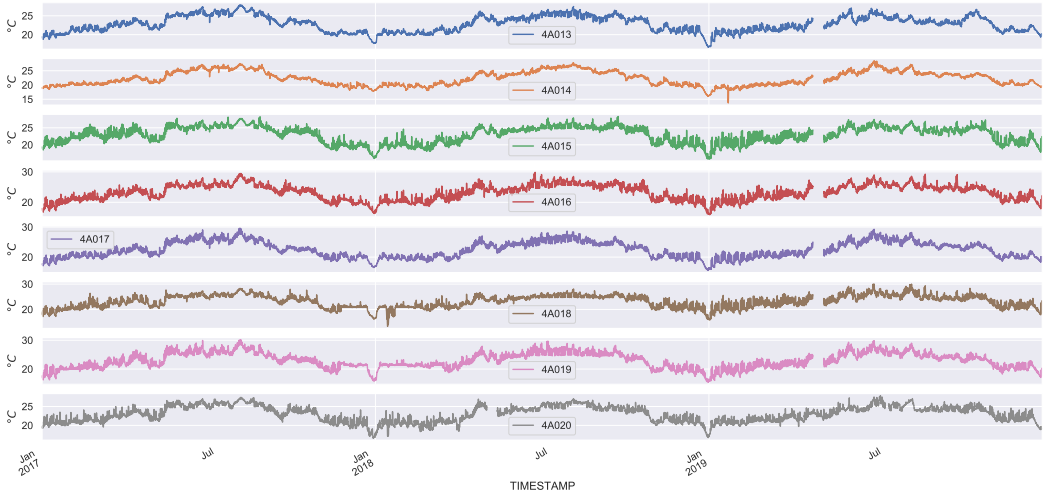


Figure 19: Overview of the measured indoor temperatures

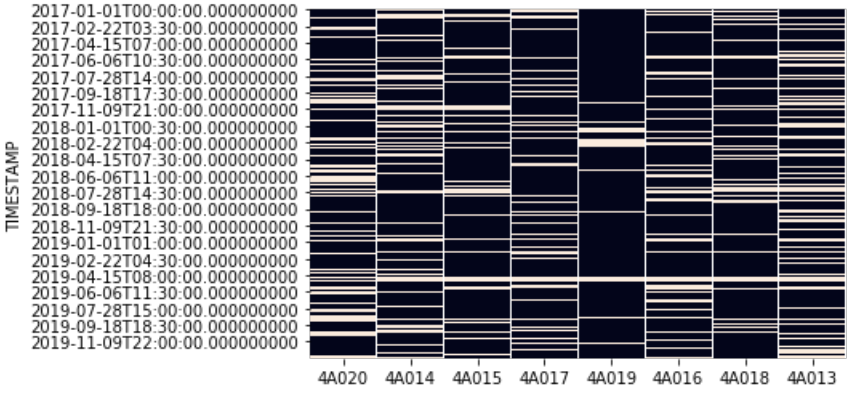


Figure 20: Overview of the missing values in the time series of indoor temperatures

A.2. Overview of the exogenous variables

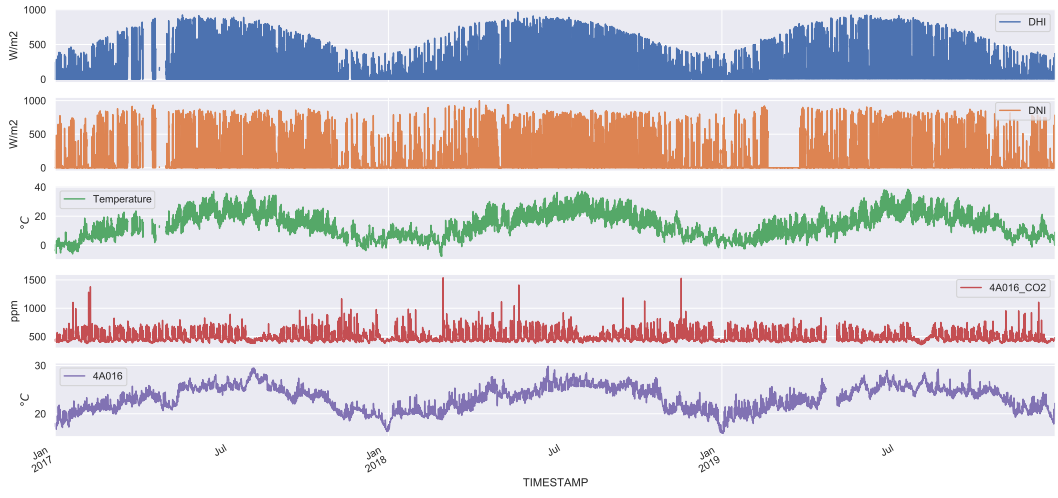


Figure 21: Overview of the solar radiation, outdoor temperature and CO2 in room 4A016

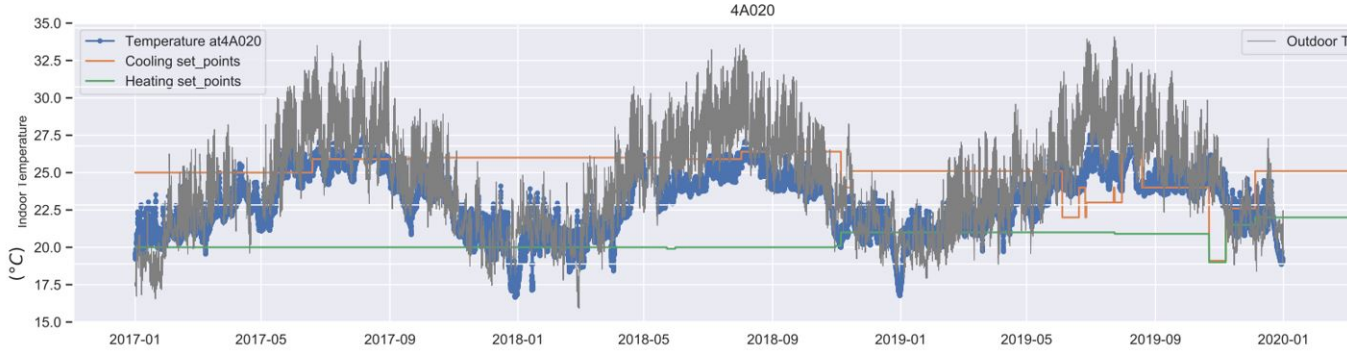


Figure 22: Overview of the room temperature set points in room 4A020

A.3. Scatter plot of the variables

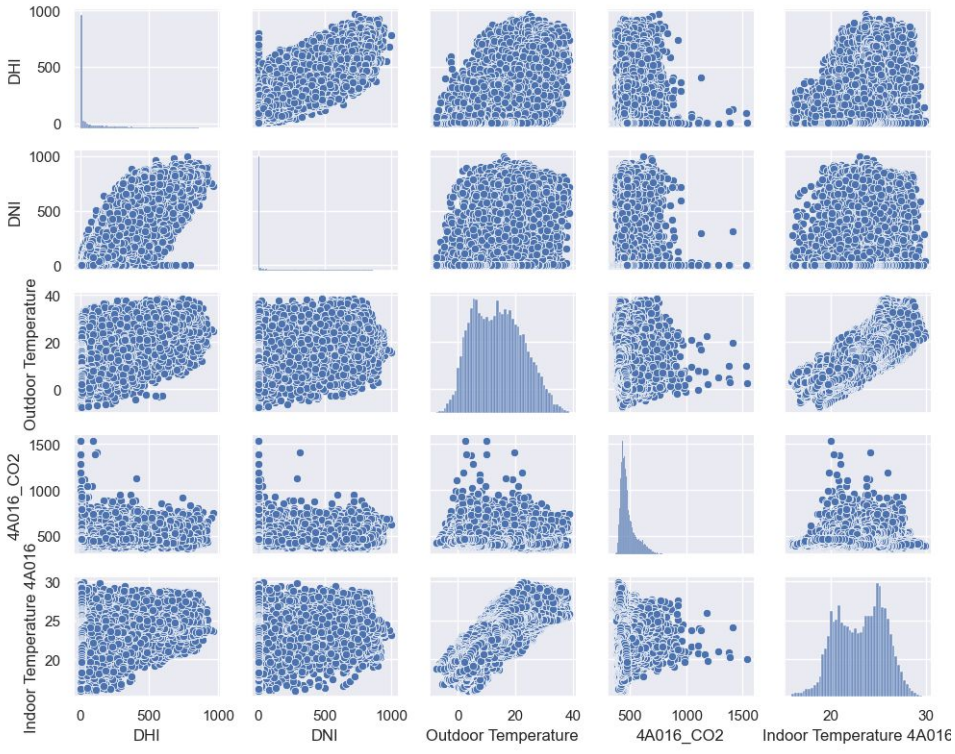


Figure 23: Scatter plot of the measured variables.

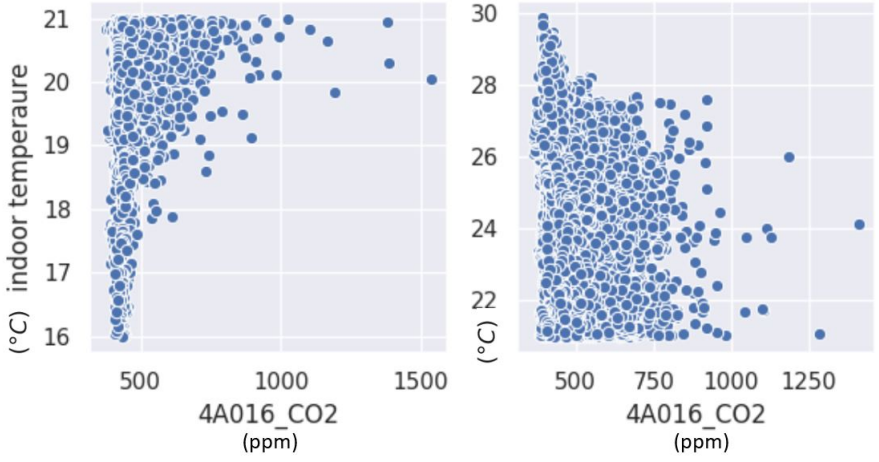
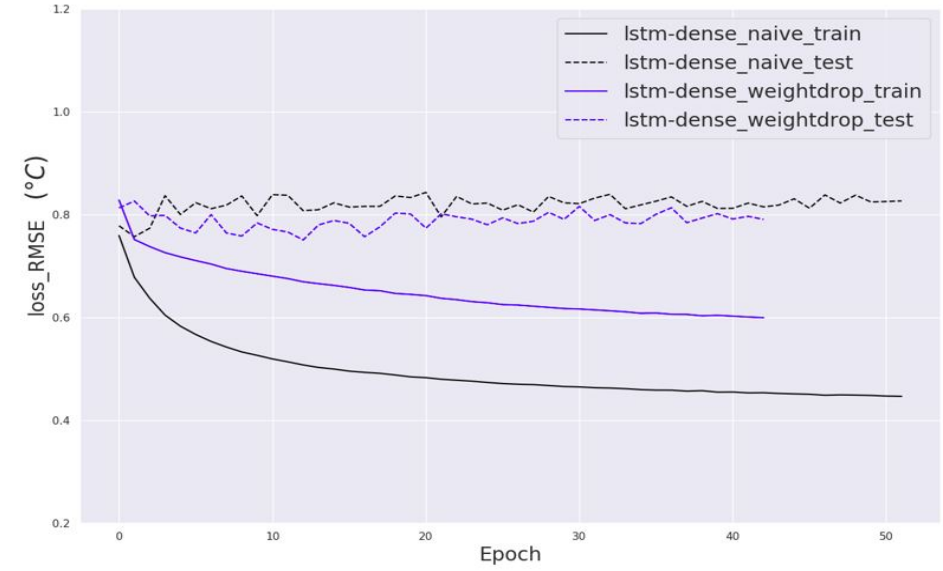
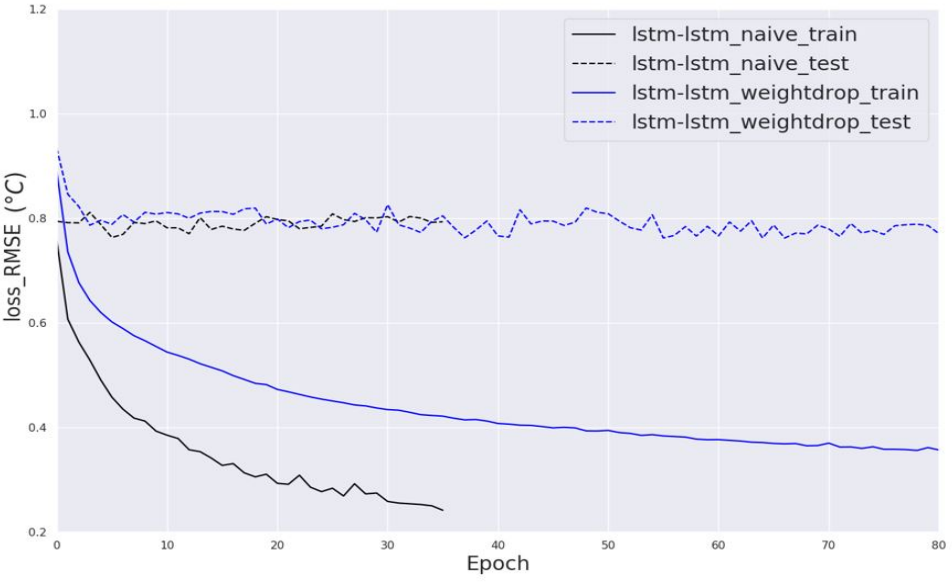


Figure 24: Scatter plot of CO2 and indoor temperature of room 4A016

A.4. Loss curve of two different dropout techniques for short-term forecasting (7d-ahead)



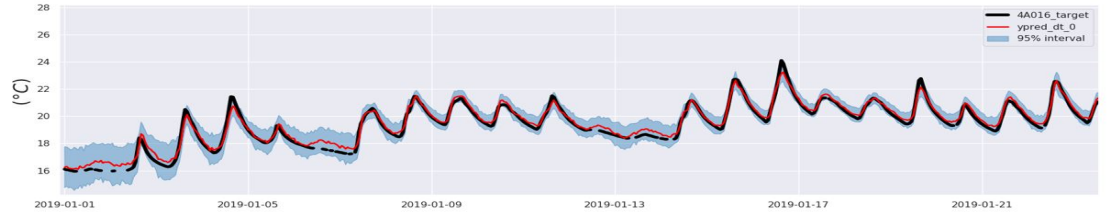
(a) LSTM-dense



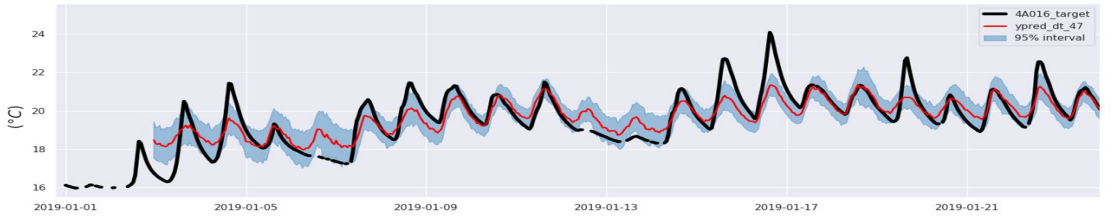
(b) LSTM-LSTM

Figure 25: training and validation loss of two different seq2seq model for 7d-ahead forecasting, with the two dropout schemes.

A.5. Prediction interval of selected periods of the year

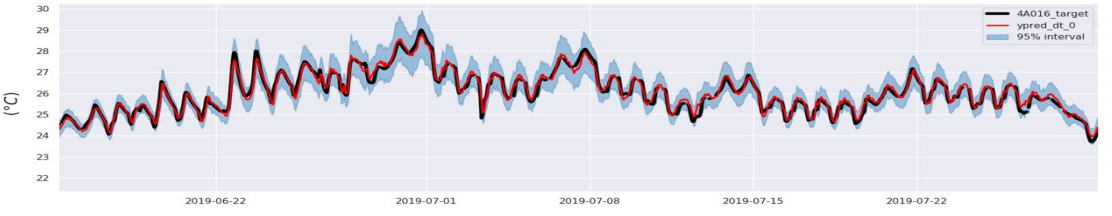


(a) hour_ahead = 1

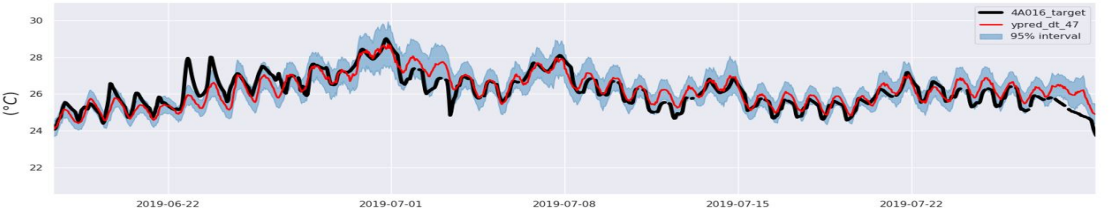


(b) hour_ahead = 48

Figure 26: Prediction interval during January

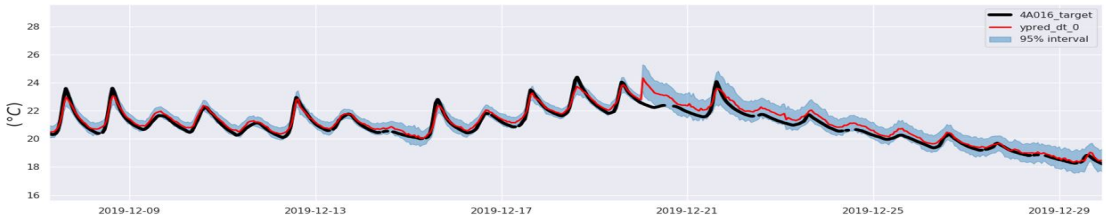


(a) hour_ahead = 1

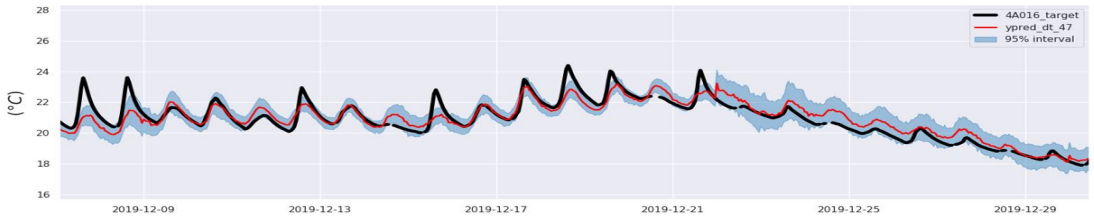


(b) hour_ahead = 48

Figure 27: Prediction interval during summer



(a) hour_ahead = 1



(b) hour_ahead = 48

Figure 28: Prediction interval during December

References

[1] A. Andreï, M. Baudry, S. Beck, A. Foussard, R. Laghouati, J. Lauverjat, P. Lévy, E. Martial, E. Misak, C. Phan, et al., Bilan énergétique de la france pour 2018 (2020).

[2] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, P. Chatellier, Modeling and forecasting building energy consumption: A review of data-driven techniques, Sustainable Cities and Society 48 (2019) 101533.

[3] T. Lu, M. Viljanen, Prediction of indoor temperature and relative humidity using neural network models: model comparison, Neural Computing and Applications 18 (2009) 345.

[4] J. E. Braun, N. Chaturvedi, An inverse gray-box model for transient building load prediction, HVAC&R Research 8 (2002) 73–99.

[5] K. Jeong, C. Koo, T. Hong, An estimation model for determining the annual energy cost budget in educational facilities using sarima (seasonal autoregressive integrated moving average) and ann (artificial neural network), Energy 71 (2014) 71–79.

[6] S. Arora, J. W. Taylor, Short-term forecasting of anomalous load using rule-based triple seasonal methods, IEEE transactions on Power Systems 28 (2013) 3235–3242.

[7] F. Zhang, C. Deb, S. E. Lee, J. Yang, K. W. Shah, Time series forecasting for building energy consumption using weighted support vector regression with differential evolution optimization technique, Energy and Buildings 126 (2016) 94–103.

[8] J. Massana, C. Pous, L. Burgas, J. Melendez, J. Colomer, Short-term load forecasting in a non-residential building contrasting models and attributes, Energy and Buildings 92 (2015) 322–330.

[9] F. Smarra, A. Jain, T. de Rubeis, D. Ambrosini, A. D'Innocenzo, R. Mangharam, Data-driven model predictive control using random forests for building energy optimization and climate control, Applied energy 226 (2018) 1252–1272.

[10] Y. Huang, H. Miles, P. Zhang, A sequential modelling approach for indoor temperature prediction and heating control in smart buildings, arXiv preprint arXiv:2009.09847 (2020).

[11] O. Ogunmolu, X. Gu, S. Jiang, N. Gans, Nonlinear systems identification using deep dynamic neural networks, arXiv preprint arXiv:1610.01439 (2016).

[12] P. Ferreira, A. Ruano, S. Silva, E. Conceicao, Neural networks based predictive control for thermal comfort and energy savings in public buildings, Energy and buildings 55 (2012) 238–251.

[13] L. Magnier, F. Haghighat, Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network, Building and Environment 45 (2010) 739–746.

[14] N. Attoue, I. Shahrour, R. Younes, Smart building: Use of the artificial neural network approach for indoor temperature forecasting, Energies 11 (2018) 395.

[15] Y. Zeng, Z. Zhang, A. Kusiak, Predictive modeling and optimization of a multi-zone hvac system with data mining and firefly algorithms, Energy 86 (2015) 393–402.

[16] Z. Afroz, T. Urmee, G. Shafiullah, G. Higgins, Real-time prediction model for indoor temperature in a commercial building, Applied energy 231 (2018) 29–53.

[17] G. Mustafaraj, G. Lowry, J. Chen, Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office, Energy and Buildings 43 (2011) 1452–1460.

[18] N. Laptev, J. Yosinski, L. E. Li, S. Smyl, Time-series extreme event forecasting with neural networks at uber, in: International Conference on Machine Learning, volume 34, 2017, pp. 1–5.

[19] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, arXiv preprint arXiv:1704.02971 (2017).

[20] G. Platt, J. Li, R. Li, G. Poulton, G. James, J. Wall, Adaptive hvac zone modeling for sustainable buildings, Energy and Buildings 42 (2010) 412–421.
[21] S. Smyl, A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting, International Journal of Forecasting 36 (2020) 75–85.
[22] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting 36 (2020) 1181–1191.
[23] L. Zhu, N. Laptev, Deep and confident prediction for time series at uber, in: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2017, pp. 103–110.
[24] L. Mba, P. Meukam, A. Kemajou, Application of artificial neural network for predicting hourly indoor air temperature and relative humidity in modern building in humid region, Energy and Buildings 121 (2016) 32–42.
[25] Z. Wang, T. Hong, M. A. Piette, Building thermal load prediction through shallow machine learning and deep learning, Applied Energy 263 (2020) 114683.
[26] C. Xu, H. Chen, J. Wang, Y. Guo, Y. Yuan, Improving prediction performance for indoor temperature in public buildings based on a novel deep learning method, Building and Environment 148 (2019) 128–135.
[27] S. Muzaffar, A. Afshari, Short-term load forecasts using lstm networks, Energy Procedia 158 (2019) 2922–2927.
[28] M. W. Ahmad, M. Mourshed, Y. Rezgui, Trees vs neurons: Comparison between random forest and ann for high-resolution prediction of building energy consumption, Energy and Buildings 147 (2017) 77–89.
[29] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and machine learning forecasting methods: Concerns and ways forward, PloS one 13 (2018) e0194889.
[30] R. Weron, Electricity price forecasting: A review of the state-of-the-art with a look into the future, International journal of forecasting 30 (2014) 1030–1081.
[31] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, B. Seaman, Sales demand forecast in e-commerce using a long short-term memory neural network methodology, in: International Conference on Neural Information Processing, Springer, 2019, pp. 462–474.
[32] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306.
[33] R. Wen, K. Torkkola, B. Narayanaswamy, D. Madeka, A multi-horizon quantile recurrent forecaster, arXiv preprint arXiv:1711.11053 (2017).
[34] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.
[35] T.-Y. Kim, S.-B. Cho, Predicting residential energy consumption using cnn-lstm neural networks, Energy 182 (2019) 72–81.
[36] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, arXiv preprint arXiv:1909.00590 (2019).
[37] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[38] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv preprint arXiv:1409.2329 (2014).
[39] S. Merity, N. S. Keskar, R. Socher, Regularizing and optimizing lstm language models, arXiv preprint arXiv:1708.02182 (2017).
[40] Y. Gal, Z. Ghahramani, A theoretically grounded application of dropout in recurrent neural networks, in: Advances in neural information processing systems, 2016, pp. 1019–1027.
[41] R. J. Hyndman, A. B. Koehler, Another look at measures of forecast accuracy, International journal of forecasting 22 (2006) 679–688.
[42] A. Khosravi, S. Nahavandi, D. Creighton, A. F. Atiya, Comprehensive review of neural network-based prediction intervals and new advances, IEEE Transactions on neural networks 22 (2011) 1341–1356.
[43] R. J. Hyndman, A. B. Koehler, R. D. Snyder, S. Grose, A state space framework for automatic forecasting using exponential smoothing methods, International Journal of forecasting 18 (2002) 439–454.