CNN-LSTM architecture for predictive indoor temperature modeling

CNN-LSTM Architecture for Predictive Indoor Temperature Modeling

Furkan Elmaz[a,*], Reinout Eyckerman[a], Wim Casteels[a], Steven Latré[a], Peter Hellinckx[a]

[a] *University of Antwerp - imec, IDLab - Faculty of Applied Engineering, Sint-Pietersvliet 7, 2000 Antwerp, Belgium*

Abstract

Indoor temperature modeling is a crucial part of achieving higher efficiency Heating, Ventilation and Air Conditioning (HVAC) systems. Completely data-driven black-box approaches have been an attractive way to develop such models due to their unique feature of not requiring detailed knowledge about the target zone. Neural network-based approaches have been taken in the literature with training models on raw data. But the noisy and non-linear nature of the problem remained as a bottleneck for the predictors, especially in long prediction horizons. In this paper, we introduce a Convolutional Neural Networks - Long Short Term Memory (CNN-LSTM) architecture to combine the exceptional feature extraction trait of the convolutional layers with the Long Short Term Memory (LSTM)'s capability of learning sequential dependencies. We experimentally collected a data set and developed Multi-Layer Perceptron (MLP), LSTM and CNN-LSTM prediction models. Models are evaluated and compared in 1-30-60-120 minutes horizons with a closed-loop prediction scheme. CNN-LSTM was able to outperform other employed methods in all prediction horizons and showed stronger robustness against the error accumulation. It managed to predict room temperature with R^2 >0.9 in a 120-min prediction horizon.

Keywords: HVAC, indoor temperature modeling, CNN-LSTM, LSTM, MLP, closed-loop prediction, black-box modeling

1. Introduction

HVAC has been the leading energy-consuming system in the buildings. 73.3% of the total energy (around 200.8 millions of tonnes of oil equivalent, Mtoe) consumption in the European Union (EU) industry accounts for heating and cooling [1]. Moreover, it is estimated that 10% to 20% of the total energy consumption in the developed countries is due to HVAC systems [2]. Thus, with the increasing awareness about the carbon footprint and depleting fossil-based resources, the development of higher-efficiency HVAC systems has become an inevitable necessity in order to reduce global energy consumption.

One of the most crucial steps to achieve this goal is the development of accurate simulation models [3]. An accurate simulation model enables the integration and validation of novel ideas without practical concerns. Since it is not feasible nor safe to implement experimental ideas to the actual buildings, simulation models play an important role in the advancement of more efficient HVAC designs. However, creating such models is not a straightforward task. HVAC systems have highly non-linear dynamics and, in addition, thermal characteristics are strongly affected by external factors such as outside temperature [4]. There has been a significant effort in the literature to develop accurate simulation models. Even though there are several different simulation objectives for different levels of HVAC systems [3], we will focus on indoor temperature modeling as the scope of this paper.

The indoor temperature modeling approaches in the literature can be categorized into three, i.e., white-box, gray-box and black-box modeling. White-box models which also known as physics-based models utilize mathematical representations of the elements based on laws of physics to develop predictive models [5]. The major shortcomings of this approach are that it requires detailed knowledge about the building/zone that one aims to model and the equations in the model usually include assumptions that do not necessarily reflect actual behavior [6].Gray-box modeling aims to narrow the gap between actual building and white-box models by introducing data-driven techniques. The most common approach is to start with a white-box model and estimate model parameters with actual building data and by using data-driven techniques [7]. Even though gray-box models showed promising results in the literature, especially compared to pure white-box models, the requirement of significant prior knowledge about the target zone and mathematical assumptions made in the models still have been a bottleneck to reach exceptionally accurate models [8].

Lastly, black-box modeling caught significant attention from the literature. One of the defining attribute

*Corresponding author

Email addresses: furkan.elmaz@uantwerpen.be (Furkan Elmaz), reinout.eyckerman@uantwerpen.be (Reinout Eyckerman), wim.casteels@uantwerpen.be (Wim Casteels), steven.latre@uantwerpen.be (Steven Latré), peter.hellinckx@uantwerpen.be (Peter Hellinckx)

List of Acronyms

HVAC . . . . Heating, Ventilation and Air Conditioning
CNN-LSTM. . . . . . Convolutional Neural Networks - Long Short Term Memory
LSTM . . . . Long Short Term Memory
CNN . . . . Convolutional Neural Networks
NNARX . . . . Neural Network-based Autoregressive Model with Exogenous Inputs
ANN . . . . Artificial Neural Networks
RNN . . . . Recurrent Neural Networks
EU . . . . European Union
R^2 . . . . Coefficient of Determination
R. . . . . Coefficient of Correlation
MAE . . . . Mean Absolute Error
MSE . . . . Mean Square Error
MPC . . . . Model Predictive Control
RL . . . . Reinforcement Learning
Std. . . . . Standard Deviation

of data-driven black-box modeling is the ability to model thermal dynamics without an explicit need for defining zone-spesific characteristics such as heat capacity and size. These characteristics are considered to be “baked” into the other input variables and their relationships. Although this situation leads more straightforward and convenient training of the models they also makes them prone be very specific to the zone they are trained for [9]. Since black-box models don’t require any prior knowledge about the building itself, it became an attractive way to develop simulation models. Furthermore, with advancements in machine learning and deep learning, many completely data-driven techniques are implemented for indoor thermal modeling. Especially, Artificial Neural Network (ANN) based implementations have been widely used. Attoue et al. used MLP to predict the indoor temperature [10]. The authors used outdoor temperature characteristics and utilized previous values of these features to develop a recurrent structure. They evaluated the models for 30 minutes to 4 hours prediction horizons and reported coefficient of correlation (R) of 0.9560 for 30 minutes to 0.8370 for 4 hours horizons. They also concluded that even though the proposed method is usable in short-horizon predictions, its performance decreases significantly as the prediction horizon widens. Although MLP is capable of predicting temperature dynamics, it does not inherently utilize the ”time-dependency”. Thus, recurrent type ANN, Recurrent Neural Networks (RNN), also have been used to develop indoor temperature models. Delcroix et al. implemented Neural Network-based Autoregressive Model with Exogenous Inputs (NNARX) architecture to predict indoor temperatures [11]. They used the logical state of heating and cooling systems (on/off), indoor temperatures, outside temperatures and relative occupancy as features. They reported coefficient of determination (R^2) values of 0.824. Furthermore, LSTM has been the state-of-the-art for predicting sequential data, thus, it has been used for indoor temperature prediction as well. Xu et al. proposed an LSTM architecture and they compared to other black-box modeling approaches such as Decision Trees and Support Vector Machines [12]. They reported R^2 values of 0.8985 for 5 minutes and 0.7956 for 30 minutes prediction horizons. Their architecture also outperformed the compared methods for each prediction horizon. Mtibaa et al. conducted a similar study where they proposed a sequence-to-sequence, also known as many-to-many, LSTM architecture to forecast future indoor temperatures [13]. They evaluated the models from 30 minutes to 6 hours predictions horizons and compared them to other baseline algorithms such as NNARX and many-to-one LSTM. They reported lower error scores and more stable prediction performance with the proposed architecture.

When the current state-of-the-art black-box modeling approaches for indoor temperature modeling are concerned, ANN-based techniques show reasonable predictive power. Moreover, RNN-based techniques, especially LSTM, show great capability for ”learning” the dynamics of the problem. On the other hand, it is certain that black-box modeling is not a silver bullet to develop an accurate simulation. In all of the mentioned studies, a significant drop in accuracy was observed as the prediction horizon widens. This is a surely expected behavior since models use their own prediction to forecast further into the future (closed-loop prediction), thus, prediction errors accumulate over time. But, it is not feasible to ignore a model’s long-term prediction capabilities. It manifests the stability and robustness of the model and allows us to investigate a portion of the dynamics learned by the model and how biased it is to training data [14]. Moreover, many modern controllers including Reinforcement Learning (RL) and Model Predictive Controllers (MPC) utilize forecasting to plan and optimize their control strategies [15, 16]. Thus, it is an important benchmark for the candidate simulation models.

In this paper, we propose including convolutional layers to improve the prediction capability of a black-box model for the indoor temperature prediction problem. We utilize convolutional layers as a feature extractor for the raw data and then the outputs are fed to an LSTM to generate predictions. This approach is known as CNN-LSTM in the literature. Although CNN-LSTM is mainly used in areas such as image captioning and natural language processing [17, 18], it has been also used for time-series tasks including energy consumption prediction [19] and stock market predictions [20]. These studies reported an increase in accuracy and stability as well as better generalization performance due to the feature extraction capability of the convolutional layers. To the best of our knowledge, there are no studies that employ CNN-LSTM architecture for the indoor temperature prediction problem. We experimentally collected a data set from a single room in Building Z of the University of Antwerp, then we employed MLP, LSTM and CNN-LSTM methods to create temperature prediction models and compare their performances. The models are evaluated on; 1-30-60-120 minutes prediction horizons. However, It should be noted that, 120 minutes time horizon is not sufficient for every temperature-effecting variable to be realized in the room, especially the external factors aside from HVAC system that can affect the temperature are concerned [21]. Thus, similar to other black-box modeling studies, the proposed methodology is more suitable for short-term predictions and unexpected deviations are likely to happen in longer prediction horizons (e.g. days). R^2, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used as fitness metrics. Predictive powers and stabilities of each model are discussed and compared.

2. Dataset Acquisition

The dataset used in this study was collected from a room in the University of Antwerp's Building Z. The sensors are located; inside the room about the room's thermal measurements, outside of the room for weather measurements and in the HVAC system to measure supply characteristics. The measurements are recorded with 1 minute sampling time and a total of 27349 samples (approx. 19 days) are collected for each sensor. Details about the measured variables and their statistical summaries are given in Table 1.

Table 1: Statistical summary of the dataset

Measurements	Min-Max	Mean	Std.
Motion (Binary)	0 - 1	0.07	0.255
Set Point ($^{\circ}C$)	10 - 22.1	19.89	1.59
Air Flowrate (m^3/h)	410.5 - 1247	638.33	352.56
Window (Binary)	0 - 1	0.006	0.08
Outside Temp. ($^{\circ}C$)	-6.38 - 20.713	6.11	6.46
Room Temp. ($^{\circ}C$)	19.76 - 22.44	20.97	0.84

where, Motion is a logical motion detector that produces 1 when there is a movement in the room and 0, otherwise. Set Point is the target temperature of the reheating battery which is controlled by a PID controller.

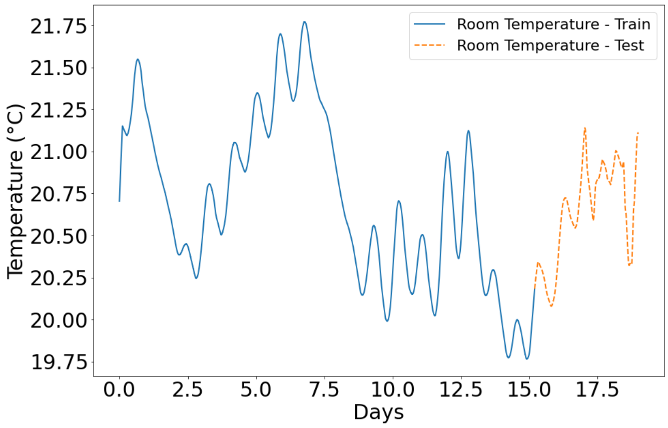


Figure 1: Plot of the room temperature variable and the train/test split

Air flowrate corresponds to the VAV flowrate. Window is the position of the single window in the room; it is 1 when open, 0 if it is closed. Outside Temp. and Room Temp. are the temperatures outside of the room and inside of the room, respectively. When the means and standard deviations of the features are concerned, Motion and Window provides very small variance in the dataset. This is due to the building and the room was generally empty regarding the COVID-19 restrictions during the measurements. Even though given measurements play crucial roles regarding the temperature dynamics of the room, there are other variables such as solar radiation and possible interactions of the occupants with the room. Although these variables are not available in our dataset, they can have significant impact on the temperature and should be considered for addition if possible to further improve prediction accuracy.

For data integration purposes, Room temperature, as the prediction target, is utilized as the output variable for the developed models. The rest of the measurements are merged to create a 27349×5 feature matrix. It should be noted that since the temperature dynamics are recurrent, previous values of the room temperature variable are also utilized as features for the proposed models. The first 75% of the data is split as a training set and the remaining portion is used as a test set. The room temperature measurements are given in Figure 1 with the legends clarifying the train-test split. Moreover, visualizations of the remaining variables are provided in the Supplementary Material.

3. Methods

Indoor temperature modeling problem can be considered as a time-series problem since the previous state of the model has an effect on the predictions [22]. This situation holds for input variables as well as for the output variable. Firstly, we employed MLP as a baseline prediction algorithm. Then, LSTM is used since its shown success for indoor temperature modeling. As the final implementation, we proposed a CNN-LSTM architecture. It should be noted that proposed neural network architectures are relatively larger compared to similar studies we discussed previously. However, we included regularization techniques in order to prevent possible overfitting scenarios. Details of the regularization implementations are given in respective sections. Moreover, since, neural networks are global function approximaters, it is safe to assume that given results are not unique to the proposed architectures in this paper and, possibly can be achieved by using different hyperparameter configurations such as number of layers/neurons with appropriate adjustments. Details of the models are given as follows:

3.1. Multi-Layer Perceptron

MLP has been considered as a "bread and butter" technique for creating a prediction model in recent years. Its' enhanced function approximation capabilities with the use of hidden layers have been successfully used in many applications, including the indoor temperature prediction problem [23]. A prediction (y_p) formulation of a single hidden layer MLP can be defined as:

$$y_p = \delta_2(H \cdot W_o + b^{(h)}) \tag{1}$$

$$H = \delta_1(X \cdot W_h + b^{(i)}) \tag{2}$$

where X is the feature matrix, W_h and W_o represent the weights from the input layer to the hidden layer and hidden layer to the output layer, respectively. δ_1 and δ_2 are the activation functions for hidden layer and output layer, respectively. Finally, the $b^{(i)}$ and $b^{(h)}$ are the bias terms for input and hidden layers. This recursive structure can be repeated a desired amount of times to "deepen" the MLP architecture.

Although the feature matrix proposed in Section 2 contains valuable information to represent the current state, thus, to predict future temperature values, using only the previous time ($t-1$) sample is not sufficient to describe the temperature dynamics as explained in Section 1. Moreover, the MLP technique does not have an inherent way of dealing with time-dependent data. Therefore, similar to other studies that utilize MLP for time-series type of data, we expand the feature matrix by adding the time-delayed versions of the features. We used 60 minutes of delay for each feature, thus, expanding the number of features in the data set from 6 (five features from the feature set, one from the delayed values of the output temperature) to 360. Inclusion of the delayed input vectors allows the model to learn different dynamics of the system, which can occur in varying periods. As stated in a review paper from Afram and Janabi-Sharif [15], 1 minute and 60 minutes time intervals are optimal to capture slow and fast-moving dynamics of an HVAC system, respectively. Thus, we include delayed features up to 60 minutes to the prediction model. It should be noted that, lower delay choices can decrease the comprehensiveness of the learned dynamics where redundant increase in delay can lead overfitting

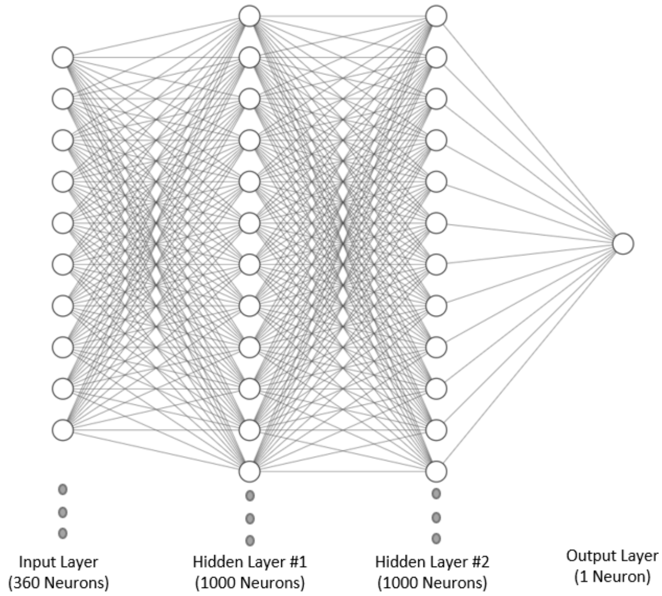


Figure 2: Illustration of the proposed MLP architecture.

as discussed in various studies that propose data-driven recurrent models [24]. For the architectural details of the MLP, we used 2 hidden layers with 1000 neurons each and ReLU activation function is used for of both them to introduce non-linearity. In order to avoid over-fitting, Dropout with the probability of 0.5 is added to the hidden layers [25]. For the training of the model, MSE is used as a cost function and Adam optimizer with weight decaying [26] is used to update the weights. The weight decay parameter, which corresponds to L2 regularization, is set to 0.01 to further prevent over-fitting scenarios. The illustration of the MLP architecture is given in Figure 2.

3.2. Long-Short Term Memory

Recurrent type of neural networks has been proposed to natively integrate the time dependency into the neural network architectures. The core idea is to expand the capability of typical ANN methods via introducing sequential structure dynamically [27]. Even though there are many techniques to achieve such functionality, LSTM has been shown to achieve exceptional success due to its capability of learning both short and long term dependencies of the problem and also designed to deal with vanishing gradient problem which most of the RNN architectures suffer from [28]. Moreover, it is a well-fit for indoor temperature modeling as well due to its inclusion of slow and fast-moving phenomenons simultaneously [15].

The main information processing units in LSTM are called "cells". These cells can be considered as more sophisticated neurons in typical MLP. A cell includes several gates to retain and regulate the flow of information over the sequence with arbitrary length. This feature enables LSTM to decide which information is useful in the long-term and the short-term. Thus, making it very suitable

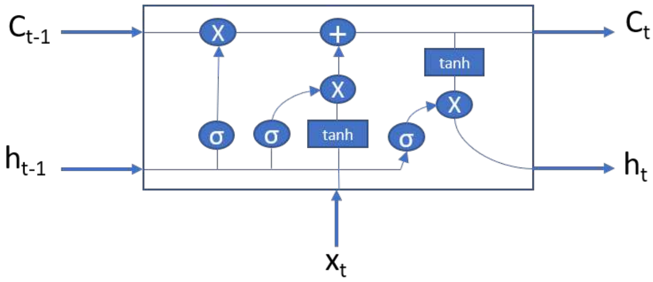


Figure 3: An LSTM Cell.

for any type of sequential-type problem. An LSTM cell can be defined as [29]:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (3)$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (4)$$

$$g_t = tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (5)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (6)$$

$$c_t = f \odot c_{t-1} + i_t \odot g_t \quad (7)$$

$$h_t = o_t \odot tanh(c_t) \quad (8)$$

where x and h are the input and the hidden state, respectively.t represent the current time step. $\odot$ is a element-wise multiplication operation (Hadamard Product) and σ is a sigmoid activation function. g_t, f_t, i_t and o_t are the cell, forget, input and output gates in the current state, respectively. W_{xy} and b_{xy} represents the learnable weights and bias terms between gates x and y. In order to have intuitive understanding about how the information flow inside of an LSTM cell, one can see the illustration in Figure 3.

LSTM cells, similar to neurons, can be connected to each other to carry temporal information and can be stacked as in layers. In this paper, we used 60 previous values of each feature. Each previous state is connected to an individual LSTM cell, thus creating a layer of 60 cells with 6 channels each. Then replica of this layer was added to deepen the LSTM network into 2 layers. Finally, we flatten the output of the LSTM layer and used a fully connected layer to obtain a single prediction.

3.3. CNN-LSTM

In many machine learning applications, feature extraction is an important step to generate meaningful information for the prediction model to enable it to make accurate predictions [30]. Time-series problems are not an exception in this regard. They include many dynamics that need to be clarified and tailored for the regression/classification model, usually with the integration of expert opinion [31]. Furthermore, feature extraction is not only a time-consuming procedure but also methodologies vastly differ from application to application [32]. Especially in recent years, researchers have been using convolution operations for automatic feature extraction rather than doing it manually. This approach is popularized with CNN architectures with ground-breaking results in computer vision problems [33]. But, its applicability is not restricted to image-type data and can be defined for time-series data as [34]:

$$Y(C_{out}) = \sum_{i=0}^{C_{in}-1} W(C_{out}, i) \star X(i) \quad (9)$$

where, Y and X are the output and the input of the convolution operation, respectively. W is the learnable parameter matrix and $\star$ is the valid cross-correlation operator. C, and L are the number of features and length of the sequence, respectively. Therefore, with the convolution operation, one can create an automated feature extraction framework and let the model learn "how" to extract features via optimizing the weights (W).

For the specific problem we focus on in this paper, the C variable corresponds to the number of features, 6, and L selected as 60 (as in the LSTM architecture). Two convolutional layers are stacked with the number of kernels of 64 and 128, and kernel sizes of 32 and 16, respectively. A stride of 1 was used in both layers and max pooling is applied at the end of each convolutional layer. All convolutional operations are implemented with a TimeDistributed layer to preserve temporal dimensions [35]. Therefore the convolution part of the proposed architecture is completed. After utilizing the spatial feature extraction capability of the convolutional layers, we introduced the LSTM part for the temporal progression of the data. The same LSTM architecture given in the previous section has been used to create the predictor including the flatten and fully connected layer. Thus, we were able to combine the spatiality feature extraction of CNN with LSTM's capability of understanding dependencies in sequential data. Final cascaded architecture is illustrated in Figure 4.

3.4. Fitness Metrics

In order to evaluate and compare the performances of the implemented methods, we employed R^2, MSE and MAE error metrics. They can be defined as:

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2}\right] \quad (10)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m}} \quad (11)$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \quad (12)$$

where, $\tilde{y}$ and y are the model predictions and the actual output, respectively. $\overline{y}$ is the output mean and m is the number of samples.

R^2 has been widely used in the literature about any type of regression type prediction tasks including indoor temperature prediction. Due to its mean-normalized definition, the output of this metric lays between 0 and 1

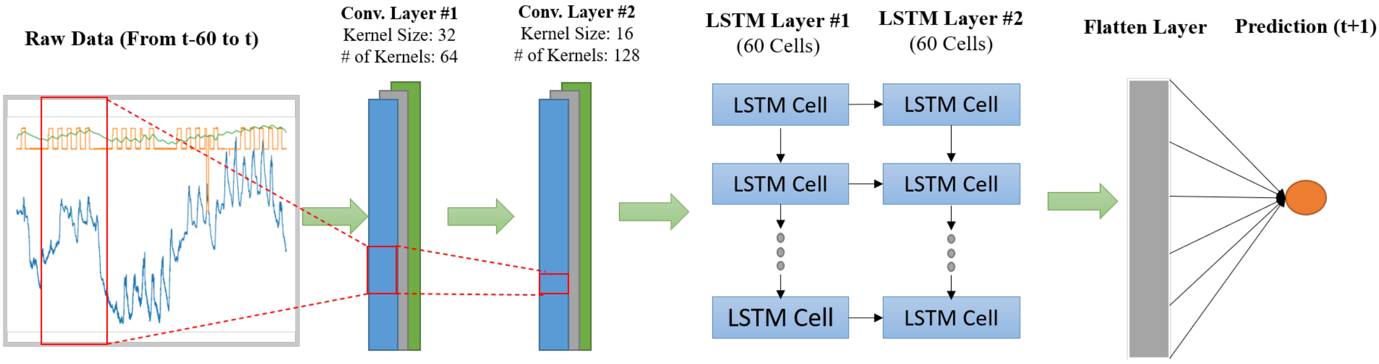


Figure 4: Proposed CNN-LSTM architecture, MaxPooling layers are not included in the illustration for clarity purposes.

where 1 corresponds to a perfect prediction. Since $RMSE$ is calculated by squaring the error term, it is highly affected by extreme values which gives us the opportunity of evaluating outlier predictions and their effects. MAE is another important metric to directly compare the performances without giving additional attention to the extreme errors. Moreover, since the differences between the actual temperature and prediction can easily lay between 0 and 1, these values tend to be ignored in $RMSE$ calculation but highly present in MAE. Thus, we employed these three fitness metrics to conduct a more in-depth analysis and interpretation of the results.

4. Results

The three different neural network architectures we employed in this paper, i.e., MLP, CNN and CNN-LSTM have been used to create prediction models for indoor temperature modeling. Although the input sizes differed depending on the architecture, all models utilized 60 min of historical data of all features and the output, thus, they had access to the same information. All models trained to predict a single variable which is the room temperature at the next time step. All of the performance evaluations are conducted on the test set which none of the models have encountered during the training.

Firstly, the models are evaluated in 1 min prediction horizon, also known as the one-step-ahead prediction. It generally shows the model's basic generalization capability and whether the model is over-fitted to the training data. But excellent one-step-ahead prediction performance is not enough to show whether the model can be used as a simulator since actual use cases of such simulators usually require relatively long-term predictions, thus, many step-ahead predictions [36]. In order to assess each model's both short and long-term prediction capabilities, we also used 30, 60 and 120-minute prediction horizons similar to the other studies in the literature. The prediction with a certain horizon corresponds to the closed-loop prediction mechanism. The model uses its own output as the feature for the predictions in further time-steps. An illustration of the used closed-loop prediction scheme is given in Figure 5.

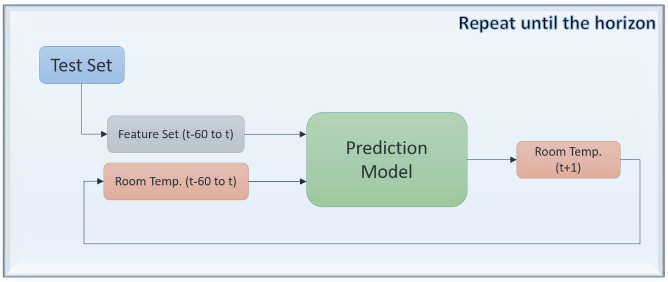


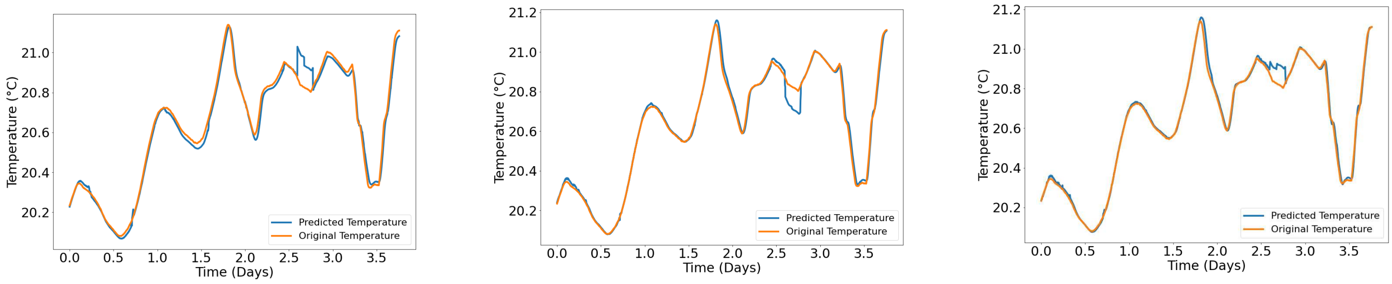
Figure 5: Illustration of the Closed-Loop Prediction Mechanism

Since the room temperature has a highly recurrent dynamic, previous values of the variable have a dramatic effect on future predictions. Thus, performance decrease with wider prediction horizons is inevitable because any deviation between a prediction and original value will accumulate to further time-steps which creates a "snowball" error effect [37]. Therefore, long-term closed-loop predictions provide us a solid way to determine a model's stability and robustness along with the prediction capability.

We evaluated the performance of the proposed models on the whole test set with repeated horizon-based predictions. In other words, the schematic given in Figure 5 has been repeated for # of datapoints the test set/prediction horizon times to cover all of the test set. Thus, the plots given below can be considered as concatenated combinations of the results. Predicted vs original temperature plots are given in Figure 6, Figure 7, Figure 8, Figure 9 for 1, 30, 60 and 120 minutes prediction horizons, respectively. R^2, RMSE and MAE metrics are calculated between the original values and predictions for the test set. These metrics are given in Table 2.

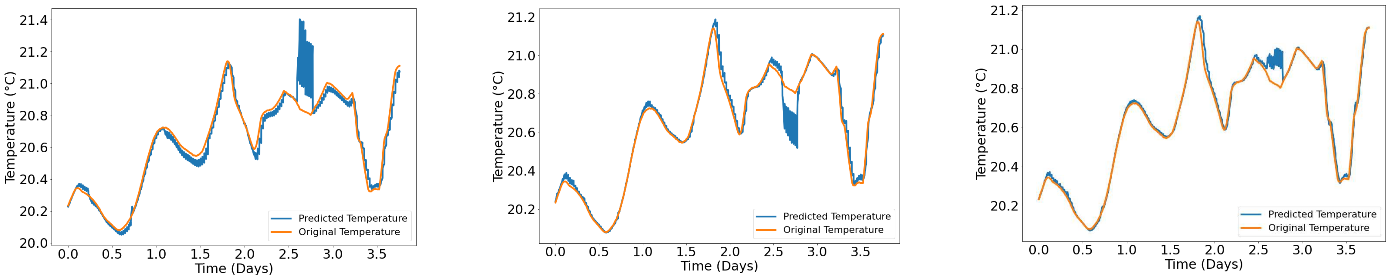
5. Discussion

When the one-step-head predictions of the models are concerned, they performed satisfactorily well and very similar. They all were able to reach R^2 >0.98 which indicates strong predictive power. On the other hand, a problematic trend between the day 2.5 and 3 can be observed in Figure 6 for all of the models. They struggled to predict



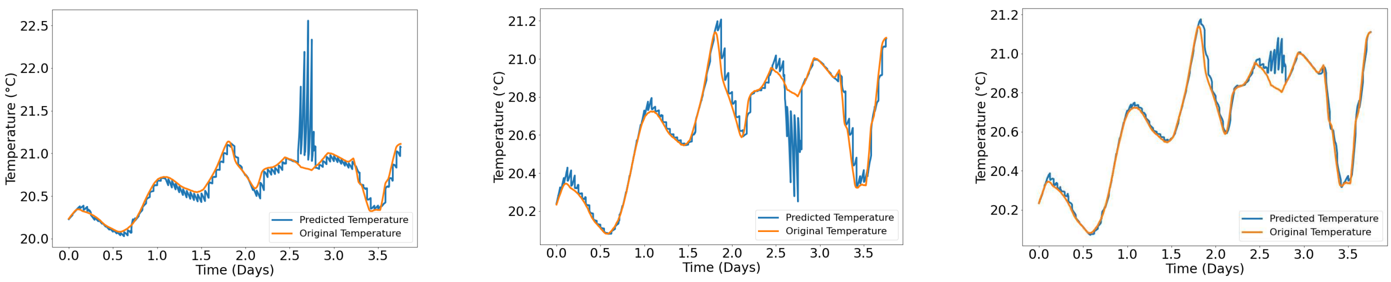
(a) MLP (b) LSTM (c) CNN-LSTM

Figure 6: Predictions vs original temperature plots for 1-min prediction horizon (one-step-ahead)



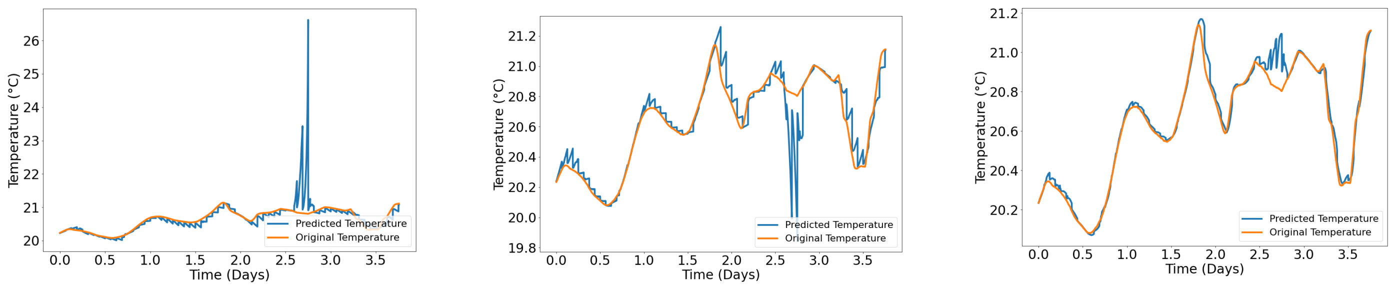
(a) MLP (b) LSTM (c) CNN-LSTM

Figure 7: Predictions vs original temperature plots for 30-min prediction horizon



(a) MLP (b) LSTM (c) CNN-LSTM

Figure 8: Predictions vs original temperature plots for 60-min prediction horizon



(a) MLP (b) LSTM (c) CNN-LSTM

Figure 9: Predictions vs original temperature plots for 120-min prediction horizon

Table 2: Performance Evaluation of the Models (RMSE and MAE values are presented in Celcius Degrees where R^2 has no unit)

Horizon	Metric	MLP	LSTM	CNN-LSTM
1 min	R^2	0.984	0.989	0.992
	RMSE	0.03	0.029	0.028
	MAE	0.025	0.019	0.015
30 min	R^2	0.935	0.972	0.984
	RMSE	0.07	0.047	0.034
	MAE	0.045	0.025	0.018
60 min	R^2	0.763	0.911	0.967
	RMSE	0.158	0.061	0.042
	MAE	0.074	0.048	0.029
120 min	R^2	0.449	0.823	0.903
	RMSE	0.325	0.121	0.052
	MAE	0.112	0.087	0.045

that specific period and the error in that region is higher than the rest of the test set. Because that period is identical for all models, it can be assumed that the temperature dynamics in that period are vastly different compared to the train set which the models trained on, in other words, models were not exposed to that specific dynamics during their training. It should also be noted that such strange behaviors can be correlated with significant outlier conditions and sensor malfunctions resulting in discontinuity and/or mislogging of the data. Although these conditions are not present in the data set we used in this study, they can also be the source of similar unexpected prediction patterns, thus, should be treated accordingly. Furthermore, it can be considered as an underwhelmingly predicted portion of the data by all models compared to the other parts of the test set. Another difference between the models is how they predicted the problematic part. MLP and CNN-LSTM architectures made "overshooting" predictions while LSTM was "undershooting". But this kind of behavior can be surely depends on the randomness during the training, such as the random seed, which does not lead to any concrete deductions.

When the prediction horizon widens, the difference between models gets more significant. As previously mentioned, closed-loop prediction performance is a strong indicator of how stable are a model's predictions. For the predictions in 30-min horizons, the MLP started performed more poorly compared to others. One can see a sharp "zig-zag" pattern starts to emerge in MLP predictions. It shows that the predictions would diverge more dramatically model will use more of its own outputs, in other words, in wider prediction horizons. This is a typical low-stability behavior where the model is not capable of handle error (which caused by MLP's previous predictions in this case) successfully. For LSTM and CNN-LSTM their errors are also amplified and the "zig-zag" pattern can be also be observed. But there are significant differences. Firstly, unlike MLP, their predictions tend to stay on top of the line and the errors are less "sharp". This can further be seen in the RMSE values in Table 2, since RMSE punishes the higher average errors much more, the dramatic difference between MLP and other methods shows that CNN-LSTM and LSTM techniques responded similarly and more robustly to the increased horizon. Furthermore, MAE difference shows the overall errors of CNN-LSTM and LSTM are considerably less than MLP.

Even though CNN-LSTM and LSTM architectures are similar in 30-min horizon performance-wise, this situation drastically changes with 60 and 120-min horizons. For the 60-min horizon, the error accumulation in the LSTM becomes predominant and it also shows signs of instability with the increased error spiking especially on the days between 2.5 and 3. CNN-LSTM also shows a decline in performance as expected, but it better copes with the error accumulation with a maximum of < 0.2 Celcius degrees as compared to LSTM's roughly 0.5 Celcius degrees. Also, the error spikes are smoother throughout all of the test set. This situation shows that the convolutional part of the CNN-LSTM architecture adds a significant value from feature extraction and the amount of important information perspective. It has been known that convolutional layers are quite capable of extracting spatial features, but also the inclusion of MaxPooling layers, it ensures that the extracted information is the most valuable and as noise-free as possible [38]. These layers force the model to understand only the most valuable features which add the ability to deal with uncertain data to the model [39]. This feature of convolutional layers is well-shown in the days between 2.5 and 3 of the test set. Lastly, MLP's performance significantly drops in 60-min horizon prediction with reaching only R^2 value of 0.763, it strongly struggles with error accumulation by reaching up to 2 Celcius degrees of error which can be considered as a red flag for any indoor temperature model. As a last prediction horizon, 120-min performances of the proposed techniques enhance the previous conclusions we drew. MLP provides predictions with unacceptable performance especially in the days between 2.5 and 3. LSTM's error spiking is much sharper with strong signs of instability and lack of robustness. CNN-LSTM, on the other hand, and managed to preserve $R^2 > 0.9$ even for such relatively wide horizon. Also between days 2.5 and 3, error tend to stay < 0.3 Celcius degrees with less of a sharpness. But it should be noted that the increase in RMSE between 60-min and 120-min horizons for CNN-LSTM is significantly larger than the increase in MAE. This shows that the error in the "problematic part" of the data, which all models failed to accurately predict starting from a 1-min horizon, is amplified much more compared to the rest of test set. This is a generalizable behavior for any type of black-box model such that; despite the precautions against instability, the variance in the data is a fundamental factor.

In order to have a deeper understanding of the performance of CNN-LSTM, we compared its performance with the state-of-the-art studies that utilize data from similar building types and uses the same prediction horizons. The

Table 3: Performance comparison between the proposed CNN-LSTM architecture (*) and other studies in the literature

Horizon	Technique	RMSE	MAE	R^2	Ref
30 min	MLP	0.03	0.08	-	[13]
	MLP	0.07	-	-	[10]
	LSTM	0.05	0.09	-	[13]
	LSTM	0.52	-	0.795	[12]
	NNARX	0.05	0.08	-	[13]
	CNN-LSTM	0.028	0.015	0.992	*
60 min	MLP	0.152	-	-	[10]
	MLP	0.04	0.13	-	[13]
	LSTM	0.05	0.12	-	[13]
	NNARX	0.06	0.1	-	[13]
	CNN-LSTM	0.042	0.029	0.967	*
120 min	MLP	0.33	-	-	[10]
	MLP	0.08	0.11	-	[13]
	LSTM	0.07	0.13	-	[13]
	NNARX	0.08	0.11	-	[13]
	CNN-LSTM	0.052	0.045	0.903	*

comparison table is given Table 3.

When the Table 3 is examined, a unique pattern between performance evaluations appears. CNN-LSTM architecture managed to reach significantly lower MAE values compared to other techniques in the recent literature. On the other hand, RMSE differences are much smaller and the MLP architecture proposed in Ref. [13] slightly outperformed the proposed one in the 30-min horizon. Because CNN-LSTM's MAE is significantly lower than the rest, it indicates the overall performance without giving additional attention to outliers was much better. But since our data contains a certain portion that has different dynamics compared to the rest, errors that occurred in the portion cause a dramatic increase in RMSE value. This can also be seen when the LSTM's performance in Table 2 and Table 3 are compared. Although LSTM proposed in this paper managed to get lower MAE values compared to the literature results, it got significantly outperformed when the RMSE values are concerned. CNN-LSTM managed to keep up with the state-of-the-art RMSE values despite being "handicapped" by the distribution of the data. It should also be noted that, although the literature results are obtained from similar buildings and features, various different factors also affect the prediction capability such as the employed architecture, software framework and data processing. Thus, one should take only the results from the methods developed in this paper as the ground truth comparison.

The prediction performance of the CNN-LSTM was superior compared to LSTM and MLP architectures developed in this paper. Similarly, it showed great promise for increasing the general prediction capability compared to other state-of-the-art models. CNN-LSTM was surely expected to outperform MLP due to its ability to inherently deal with time-series types of data. But its elevated performance compared to vanilla LSTM architecture shows the importance and added value of the convolutional layers. One can conclude that the spatial feature extraction capability of the convolutional layers strongly enhanced the quality of the information passed to the LSTM layers which leads to more stable and accurate predictions even under uncertainty. Thus, CNN-LSTM showed that it is a powerful type of black-box modeling method for indoor temperature prediction and poses a strong potential for other HVAC modeling tasks. However, it surely inherits the well-known limitations of black-box modeling as well. It is important to take a look at black-box modeling for indoor temperature modeling in general to define our future work.

Machine learning literature has been providing researchers many powerful tools to create prediction models without a need for prior knowledge about the building itself. Although this is a favorable feature, it also adds limitations. Most importantly, the reasonable variance in the data is essential while building a black-box model. Since most of the data from buildings are collected while the building is operational, the data range becomes quite limited due to the requirement of ensuring thermal comfort and safety of the people inside the building. Thus, the collected data, especially temperature, lays within a limited range as is the case in this study. Therefore, developed models become applicable for certain working conditions unless the data collection is performed with variance in mind which is rarely the case. As our future work, we will focus on finding a methodology to utilize white-box and gray-box models to generate high variance data and combine it with the actual building data effectively to provide a more "wide" range of information. Thus, creating still a completely data-driven CNN-LSTM model but with the increased variance and smoother error distribution.

Lastly, we would like to discuss the potential application of the proposed CNN-LSTM architecture and how its enhanced stability and predictive power compared to other black-box modeling techniques can be utilized. First obvious application is for relatively short-term simulations. Because HVAC systems in the buildings can be considered as safety-critical, it is not possible to conduct experiments on the actual system due to safety risks. Thus, predictive models enable experimental implementations and "what if" scenarios by serving as digital twin of the building. One can manipulate the controllable parameters such as temperature set points and air flow rate to observe their effects for the target zone in order to develop more energy efficient and more thermally comfortable strategies for its users. However, we believe that the major contribution of the proposed CNN-LSTM architecture is in modern controller designs. In the current state, HVAC systems, especially in industrial scale, have been generally using PID and rule-based controllers. Meanwhile, academia have been challenging these conventional approaches by proposing modern controller designs. Most notably, MPC and RL-based controllers have shown to create more sophisticated control strategies that has significant potential to decrease energy consumption over time, superior adaptation to changing conditions, higher customizability depending on the desired thermal conditions [15]. The main advantage of the mentioned modern controllers is their ability to "plan" into the future by utilizing predictive models. These predictive models allow controllers to observe consequences of their actions without interacting with the real environment and optimize their strategy to achieve desired control objective(s). The main issue of these predictive models are the computational complexity. Especially in real-time control applications, the predictive model has to provide results in short notice. Because black-box models has very low computational burden to generate predictions once they are trained, they are a perfect match as predictive models. On the other hand, the stability and robustness of the black-box models are generally the focus of the criticism against them [9]. With the increased stability and predictive capability of the CNN-LSTM architecture compared to other architectures employed in this study, it also shows great potential to be used in modern controller designs.

6. Conclusion

In this study, we proposed a black-box CNN-LSTM architecture for indoor temperature modeling and compared its performance to MLP and LSTM. The dataset is collected from a room in the University of Antwerp's Building Z and included 6 types of measurements, i.e., motion detector, set point temperature, air flowrate, window position, outside temperature and room temperature. 60 mins of previous data for each measurement are utilized while developing the models. The dataset was divided into 75% training and 25% testing sets. Firstly, we created an MLP model where each feature and its previous values are fed as separate outputs. Then an LSTM architecture is developed and the data is fed sequentially. Lastly, we proposed a CNN-LSTM model where the convolutional part of the architecture was used as a feature extractor before LSTM. Results of the models are evaluated in 1-30-60 and 120 min prediction horizons in a closed-loop prediction fashion where models used their own outputs to forecast further into the future. In 1-min horizon predictions, all models performed exceptionally well. But a certain portion of the data was "problematic" for all of the models. With the increase in prediction horizons, MLP and LSTM showed stability problems and their predictive capability diminished dramatically. This performance drop was much more present in the "problematic" part of the data. Although CNN-LSTM's performance also dropped due to accumulating error of predictions, it showed less spiking error pattern and more stability. It managed to stay $R^2 > 0.9$ in a 120-min prediction horizon. Thus, the addition of the convolutional layers is shown to add value to the model. Feature extraction and noise reduction capabilities of these layers managed to provide the most valuable information to the LSTM layers with also the help of dropout layers. Therefore, CNN-LSTM proved itself as a valuable black-box modeling approach for indoor temperature modeling. On the other hand, its limitation due to the data variance was clearly present. Even though it managed to cope with the distribution differences on the data set very robustly, it is still a limiting factor just like for all of the black-box techniques. In future work, we will focus on; solving the limitation of variance by incorporating other modeling approaches, increasing the models' reliable predictive ranges to matter of days for long term usability, thus, enabling the use of CNN-LSTM architecture to its full potential.

References

[1] T. Fleiter, J. Steinbach, M. Ragwitz, J. Dengler, B. Köhler, F. Reitze, F. Tuille, M. Hartner, L. Kranzl, S. Forthuber, et al., Mapping and analyses of the current and future (2020-2030) heating/cooling fuel deployment (fossil/renewables), European Commission, Directorate-General for Energy (2016).

[2] S. K. Alghoul, A comparative study of energy consumption for residential hvac systems using energyplus, American Journal of Mechanical and Industrial Engineering 2 (2) (2017) 98–103.

[3] A. Afram, F. Janabi-Sharifi, Review of modeling methods for hvac systems, Applied Thermal Engineering 67 (1-2) (2014) 507–519.

[4] F. Tang, A. Kusiak, X. Wei, Modeling and short-term prediction of hvac system with a clustering algorithm, Energy and Buildings 82 (2014) 310–321.

[5] D. Enescu, A review of thermal comfort models and indicators for indoor environments, Renewable and Sustainable Energy Reviews 79 (2017) 1353–1379.

[6] F. Amara, K. Agbossou, A. Cardenas, Y. Dubé, S. Kelouwani, et al., Comparison and simulation of building thermal models for effective energy management, Smart Grid and renewable energy 6 (04) (2015) 95.

[7] A. Afram, F. Janabi-Sharifi, Gray-box modeling and validation of residential hvac system for control system design, Applied Energy 137 (2015) 134–150.

[8] R. Z. Homod, Review on the hvac system modeling types and the shortcomings of their application, Journal of Energy 2013 (2013).

[9] A. Afram, F. Janabi-Sharifi, Black-box modeling of residential hvac system and comparison of gray-box and black-box modeling methods, Energy and Buildings 94 (2015) 121–149.

[10] N. Attoue, I. Shahrour, R. Younes, Smart building: Use of the artificial neural network approach for indoor temperature forecasting, Energies 11 (2) (2018) 395.

[11] B. Delcroix, J. Le Ny, M. Bernier, M. Azam, B. Qu, J.-S. Venne, Autoregressive neural networks with exogenous variables for indoor temperature prediction in buildings, in: Building Simulation, Springer, 2020, pp. 1–14.

[12] 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.

[13] F. Mtibaa, K.-K. Nguyen, M. Azam, A. Papachristou, J.-S. Venne, M. Cheriet, Lstm-based indoor air temperature prediction framework for hvac systems in smart buildings, Neural Computing and Applications 32 (23) (2020) 17569–17585.

[14] 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.

[15] A. Afram, F. Janabi-Sharifi, Theory and applications of hvac control systems–a review of model predictive control (mpc), Building and Environment 72 (2014) 343–355.

[16] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building hvac control, in: Proceedings of the 54th annual design automation conference 2017, 2017, pp. 1–6.

[17] K. Yoshida, M. Minoguchi, K. Wani, A. Nakamura, H. Kataoka, Neural joking machine: Humorous image captioning, arXiv preprint arXiv:1805.11850 (2018).

[18] A. M. Alayba, V. Palade, M. England, R. Iqbal, A combined cnn and lstm model for arabic sentiment analysis, in: International cross-domain conference for machine learning and knowledge extraction, Springer, 2018, pp. 179–191.

[19] T.-Y. Kim, S.-B. Cho, Predicting residential energy consumption using cnn-lstm neural networks, Energy 182 (2019) 72–81.

[20] S. Liu, C. Zhang, J. Ma, Cnn-lstm neural network model for quantitative strategy analysis in stock markets, in: International Conference on Neural Information Processing, Springer, 2017, pp. 198–206.

[21] N. Ma, D. Aviv, H. Guo, W. W. Braham, Measuring the right factors: A review of variables and models for thermal comfort and indoor air quality, Renewable and Sustainable Energy Reviews 135 (2021) 110436.

[22] A. Rajith, S. Soki, M. Hiroshi, Real-time optimized hvac control system on top of an iot framework, in: 2018 Third international conference on fog and mobile edge computing (FMEC), IEEE, 2018, pp. 181–186.

[23] S. Alawadi, D. Mera, M. Fernández-Delgado, F. Alkhabbas, C. M. Olsson, P. Davidsson, A comparison of machine learning algorithms for forecasting indoor temperature in smart buildings, Energy Systems (2020) 1–17.

[24] G. Abbas, M. Nawaz, F. Kamran, Performance comparison of narx & rnn-lstm neural networks for lifepo4 battery state of charge estimation, in: 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), IEEE, 2019, pp. 463–468.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[26] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[27] D. Mandic, J. Chambers, Recurrent neural networks for prediction: learning algorithms, architectures and stability, Wiley, 2001.

[28] D. Arpit, B. Kanuparthi, G. Kerg, N. R. Ke, I. Mitliagkas, Y. Bengio, h-detach: Modifying the lstm gradient towards better optimization, arXiv preprint arXiv:1810.03023 (2018).

[29] H. Sak, A. Senior, F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, arXiv preprint arXiv:1402.1128 (2014).

[30] S. Khalid, T. Khalil, S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, in: 2014 science and information conference, IEEE, 2014, pp. 372–378.

[31] F. Ongenae, S. Van Looy, D. Verstraeten, T. Verplancke, D. Benoit, F. De Turck, T. Dhaene, B. Schrauwen, J. Decruyenaere, Time series classification for the prediction of dialysis in critically ill patients using echo statenetworks, Engineering Applications of Artificial Intelligence 26 (3) (2013) 984–996.

[32] Z. M. Hira, D. F. Gillies, A review of feature selection and feature extraction methods applied on microarray data, Advances in bioinformatics 2015 (2015).

[33] M. Jogin, M. Madhulika, G. Divya, R. Meghana, S. Apoorva, et al., Feature extraction using convolution neural networks (cnn) and deep learning, in: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, 2018, pp. 2319–2323.

[34] D. Li, J. Zhang, Q. Zhang, X. Wei, Classification of ecg signals based on 1d convolution neural network, in: 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), IEEE, 2017, pp. 1–6.

[35] A. Chawla, B. Lee, S. Fallon, P. Jacob, Host based intrusion detection system with combined cnn/rnn model, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 149–158.

[36] M. Gustin, R. S. McLeod, K. J. Lomas, Forecasting indoor temperatures during heatwaves using time series models, Building and Environment 143 (2018) 727–739.

[37] H. Cheng, P.-N. Tan, J. Gao, J. Scripps, Multistep-ahead time series prediction, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2006, pp. 765–774.

[38] P. Satti, N. Sharma, B. Garg, Min-max average pooling based filter for impulse noise removal, IEEE Signal Processing Letters 27 (2020) 1475–1479.

[39] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L. M. Gambardella, Max-pooling convolutional neural networks for vision-based hand gesture recognition, in: 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), IEEE, 2011, pp. 342–347.