2021 23rd International Conference on Process Control (PC)
June 1–4, 2021, Štrbské Pleso, Slovakia

Prediction of User-defined Room Temperature Setpoints by LSTM Neural Networks

Nikica Perić[a,*], Hrvoje Novak[b], Filip Rukavina[c] and Vinko Lešić[d]
[a,b,c,d]*Laboratory for Renewable Energy Systems*
University of Zagreb, Faculty of Electrical Engineering and Computing
Zagreb, Croatia,
[a,*]nikica.peric@fer.hr, [b]hrvoje.novak@fer.hr, [c]filip.rukavina@fer.hr, [d]vinko.lesic@fer.hr

***Abstract*—Buildings are one of the largest consumers of energy in the world and, as such, an opportunity for significant energy savings. Modelling of user behavior is a prerequisite for predicting the building energy consumption. It offers a possibility to adapt the building operation to best suit the user requirements while achieving minimal energy consumption through room air conditioning. This paper presents a deep learning model for prediction of an individual room temperature setpoint provided by the user. Model learning is conducted on the available dataset consisting of external weather and internal climate measurements. External weather conditions include the air temperature, relative humidity and direct, diffuse and global solar irradiation. The internal conditions refer to four offices for which the air temperature, the mode of operation of the fan coil controller and the temperature setpoints are available. The model is based on a neural network with Long Short-Term Memory. The highest accuracy is achieved with the model developed for the inputs of historical setpoints, system operation, day of the week and relative humidity. Presented results show high model accuracy, applicable for real-time model-based predictive building operation.**

***Index Terms*—user behavior modelling, room temperature reference prediction, deep learning, LSTM, neural networks**

I. Introduction

Accountnig for one third of the world's total energy consumption, buildings sector is one of the biggest consumers and a suitable area for large energy savings [1]. A significant opportunity lies in reduction of consumption in the building heating, ventilation and air conditioning (HVAC) system. From a technical point of view, there are two approaches of controlling HVAC systems: model-based physical methods, such as model predictive control, and black box methods, such as artificial neural networks (NN) [2]. Traditional methods of the building heating and cooling omit the habits and priorities of the individual user, which are recently becoming emphasized in the development of office climate conditioning control algorithms to increase the productivity of the occupants. Therefore, modelling of user behavior provides an opportunity to improve the ambience of the room, as well as to predict consumption and possible optimization of air conditioning of the room and the whole building [3].

This paper presents a Long Short-Term Memory (LSTM) model that captures user behavior by predicting the desired room temperature setpoint at a period of one year at different weather conditions, times of the day, seasons, and working and non-working days. The training of the model is conducted on a dataset containing external weather conditions and internal climate measurements, and a set of artificially calculated variables designed to improve the model accuracy. There are two basic approaches when using neural networks. The first is an approach in which a number of input variables is used to estimate the output that is directly related to the inputs. The second method is based on the time dependence between past and future observed variables, in this paper the temperature setpoint [4]. In [5] LSTM is used to predict the setpoint, but does not use external weather measurements such as relative humidity and solar irradiance. In [6] LSTM is used for power management. LSTMs are often used in so-called deep learning to predict time series in recent years. Other neural network types for temperature setpoint prediction are presented in [7] and [8]. Presented model predicts temperature setpoint in real time 24 hours in advance, and the accuracy of the found LSTM model is compared to the accuracy of the rule based model. Generated temperature setpoint predictions are suitable to be utilized in predictive control, such as [9] where the predicted setpoint trends are used to aid in the optimization of the buildings heating/cooling system. Such information also offers insights into the daily heating/cooling building energy demands and offers a possibility to plan the system operation in advance (e.g. preheating/precooling).

In this paper, the dataset is described in Section II. Then the training methodology is described in Section III, which shows how LSTM and its hyperparameters and relevant inputs are selected. In Section IV, a decision for final input variables selection is elaborated. The results showing a comparison of the LSTM model are shown in Section V. In Section VI, the conclusion is given.

II. Dataset

In this paper, LSTM models are implemented that predict temperature setpoints ($T_{setpoint}$ below) for four rooms located at the Faculty of Electrical Engineering and Computing in Zagreb. Data analysis in this section and LSTM model tuning are presented for one of the four rooms, and the results section shows the obtained model accuracy for all four rooms. The data used for network learning are taken from the historical database, and refer to the whole year of 2019. Measurements in the database are collected with a meteorological station described in [10] and [11]. Only the data for the period when heating was enabled in the building is used to create the model, which for 2019 was from January to May and from October to December. The cooling season lasts only from June to September, often lacks data due to holidays and database downtime, and the frequent manual mode where the setpoint is often impossible to be extracted. The unavailable setpoints may be observed in Fig. 1.

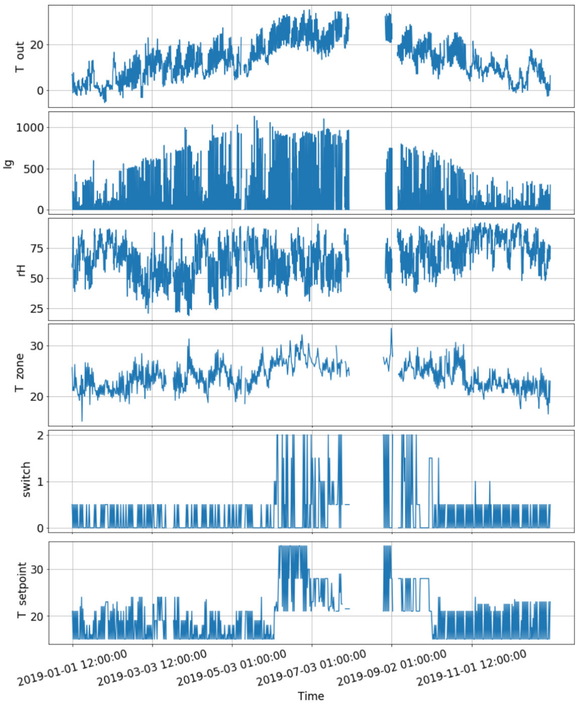

Fig. 1. Display of variables related to external and internal conditions

The input data for LSTM are divided into external and internal conditions, which are archived for all rooms at a frequency of one set of measurements per minute.

External conditions contain data on the outdoor temperature T_{out} measured in °C, global solar radiation I_g, direct solar radiation I_{dir} and diffuse solar radiation I_{diff} expressed in watts per square meter (W/m^2), as well as external relative humidity rH in percentage (%). The values of T_{out}, I_g and rH for 2019 are shown in Fig. 1.

Figure 1 also shows internal conditions measurements for 1 year with a resolution of one hour, including zne temperature T_{zone}, local fan coil mode of operation $switch$, and $T_{setpoint}$. The T_{zone} is shown in °C. Variable $switch \in \{0,\ 0.5,\ 1,\ 1.5,\ 2\}$ shows the mode of operation of the fan coil controller. During the heating season, 0 means that $T_{setpoint}$ is 15 °C and fan coil is switched off, 0.5 represents the automatic mode in which $T_{setpoint}$ is in range from 18 to 24 degrees, 1 is manual in 1st fan speed, 1.5 is manual in 2nd, and 2 represents manual mode in 3rd fan speed. The user selects $T_{setpoint}$ in automatic mode, and the data from this mode are crucial for further observations. Temperature setpoint is a value set depending on the heating and cooling needs of the room when the system is operating in automatic mode. In the developed model, $T_{setpoint}$ is selected as the output to be predicted 24 hours in advance, with a frequency of 1 hour.

In addition to the obtained measurements, several variables were calculated to provide information on day of the week (day_{ow}), time in hours (t_h) and information on whether the day is working or non-working (day_w). Also, the variables $system_{on/off}$, w_{zone}, w_{all} and $rH_{improved}$ have been added and are described in the Section III.

All the listed data is stored in the considered dataset excluding the incomplete data. Thereafter, the data is divided into training, validation and test dataset in the ratio of approx 65%-20%-15% and scaled by the min-max method using:

$$x_s = \frac{(x - x_{\min}) \cdot (x_{s,\max} - x_{s,\min})}{x_{\max} - x_{\min}} - x_{s,\min}, \qquad (1)$$

where $x_{s,max}$ and $x_{s,min}$ are the limits of the scaled values, x_{max} and x_{min} are the limits of original values, x_s represents the scaled and x the original data.

III. Modelling of Desired Temperature Setpoint

Prediction of $T_{setpoint}$ defined by the user is modeled by using an LSTM neural network implemented by using the Tensorflow and Keras libraries in the Python programming language.

A. *Long Short-Term Memory*

The LSTMs is a type of recurent neural networks (RNNs) capable of learning long-term dependencies. They were proposed by Hochreiter and Schmidhuber in 1997 [12] as networks that better handle vanishing gradient problem better, when compared to other RNNs. Due to their applicability to various deep learning problems, such as speech recognition, handwriting, translation and time series, they are widely used today.

Considered LSTMs have a chain structure of cells with four layers within each cell. Each cell represents one timestamp. Neighboring cells are connected by two vectors: state of the cell (C_t) and hidden state (h_t), which helps them in learning time dependencies.

For the application of predicting the user temperature setpoints, the LSTM network is chosen because of its robustness, better resolution of the vanishing and exploding gradient, and the ability to learn larger intervals between events crucial to the neural network as opposed to classical RNNs.

B. *Creation of additional variables*

The historical data described in Section II is processed to facilitate better model accuracy. Additional variables are created based on a-priori knowledge of the modelled process: system on-off indicator ($system_{on/off}$ below), w_{zone}, w_{all} and $rH_{improved}$.

The variable $system_{on/off}$ indicates whether the system is operational ($T_{setpoint}$ from 18 °C to 24 °C) or not ($T_{setpoint}$ is fixed at 15 °C). This way, LSTM learns to better distinguish when to set $T_{setpoint}$ to 15 °C and when other variables are more important.

Furthermore, the w_{zone} is introduced as a slack variable to detect when the room temperature T_{zone} is higher than the temperature set by $T_{setpoint}$ during working hours:

$$w_{zone} = \max(T_{zone} - T_{setpoint}, 0). \tag{2}$$

This way, it is detected when the user might feel that the temperature is too high, which is why the user could lower $T_{setpoint}$ to switch off the system. The reverse situation tends to happen less often in the observed dataset and in such a situation the heating is already on, which is why the results are better without negative values of w_{zone}.

The w_{all} variable works in a similar way, in which, in addition to the room temperature, the outside temperature T_{out} and global solar radiation I_g are also used in order to capture subjective perception on what is actually the user's sense of room temperature. This variable is calculated during working hours by the formula:

$$w_{all} = a \cdot T_{zone} + b \cdot T_{out} + c \cdot I_{diff} - T_{setpoint}, \tag{3}$$

where a, b and c are arbitrary coefficients whose selection determines the influence of T_{zone}, T_{out} and I_{diff} to w_{all}. They are determined experimentally in order for w_{all} to better describe the user's subjective perception of the temperature.

The variable $rH_{improved}$ is an additional slack variable corresponding to the relative humidity rH during working hours, and during non-working hours it is saturated at a maximum value of 100. Since rH is negatively correlated with $system_{on/off}$ (-0.13), it can be concluded that rH during non-working hours (mostly night) is on average higher than during working hours. Therefore, it was decided to set $rH_{improved}$ to 100 during non-working hours, which makes it easier for the model to use information on humidity influence. Testing has also confirmed that the model responds better if $rH_{improved}$ is saturated at the highest value than e.g. at 0. Table I shows all the input variables used in the model and their brief descriptions.

TABLE I
LIST OF INPUTS USED IN THE MODEL

Input	Description
T_{setpoint}	setpoint (desired) temperature
T_{zone}	zone (room) temperature
T_{out}	outside temperature
I_{g}	global solar irradiance
I_{dir}	direct solar irradiance
I_{diff}	diffuse solar irradiance
day_{ow}	day of week $\in \{1, ..., 7\}$
day_{w}	working or non working day $\in \{0, 1\}$
$hour$	time in the day $\in \{0, ..., 23\}$
$switch$	local switch state $\in \{0, 0.5, 1, 1.5, 2\}$
$system_{\text{on/off}}$	indicator of whether the system is turned on $\in \{0, 1\}$
w_{zone}	warmth based on T_{setpoint} and T_{zone}
w_{all}	warmth based on T_{setpoint}, T_{zone}, T_{out} and I_{diff}
rH	external relative humidity
rH_{improved}	combined rH and $system_{\text{on/off}}$

C. Selection of the neural network structure

As already mentioned in this paper, the LSTM neural network was selected. The chain contains as many cells as the number of history timestamps ($history\ size$) used to predict number of future timestamps ($future\ target$). LSTM inputs are previous day measurements that predict $T_{setpoint}$ for each of the next 24 hours in a single evaluation.

The input data are in the form of a three-dimensional array where the first dimension is infinite so that the model can be constantly fed with data, and the weight factors in the network are refreshed with each $batch\ size$ of examples. The second and third dimensions of the array refer to $history\ size$ and to the number of different input variables that the network has, which together form the input matrix x_s of the scaled data. The output data are in the form of a two-dimensional array where first dimension is infinite, and the second corresponds to $history\ size$. There is no third dimension because of a single output $T_{setpoint}$.

In addition to the LSTM layers, it is possible to add other different inner layers to the neural network, e.g. the *dense layer* layer, which is a characteristic of feedforward neural networks. Also, it is common to use some of the regularization methods to make the network less prone to overfitting. Dropout [13], lasso (L1) and ridge (L2) methods were tested and the dropout method proved to be the best (error reduced by about 2%). Also, in all types of neural networks, including LSTM, one of essential items is the activation function. The activation function nonlinearly transforms the data to make it easier to handle. LSTM-s as standardized activation functions have sigmoid and tangent hyperbolic functions. The final set of the network layers was obtained below.

D. Selection of other hyperparameters

In addition to the structure of the neural network, it is also necessary to select values and calculation methods for some other parameters. Some of these parameters are: $history\ size$, $future\ target$, optimizer, $learning\ rate$, loss function, $number\ of\ epoch$, $batch\ size$, $buffer\ size$. Through network testing, different values and ways of calculating these parameters were tested, and Table II shows the best among them.

TABLE II
BEST VALUES AND METHODS FOR PARAMETERS

Hyperparameter	Value
Layers and activation functions	LSTM (50, tanh, sig) Dropout(0.1) LSTM (50, tanh, sig) Dense (24, tanh)
history size	24
future target	24
optimizer	Adam
learning rate	0.005
loss function	mean squared error
number of epochs	100 with early stopping
batch size	100
buffer size	whole train data

Neural network learning is performed through several epochs consisting of a certain number of steps ($steps\ per\ epoch$), and in each step it contains $batch\ size$ examples. One epoch represents the cycle in which all the learning data were processed. After each epoch, the value of the loss function in the training and test data are checked. For the loss function, mean squared error (MSE) was chosen, which, compared to the also frequently used mean absolute error (MAE), more often gives good or at least acceptable results, i.e. it has less large deviations from the user's needs. Among the optimizers, SGD, RMSprop, and Adam [14] were tested, and the Adam optimizer with $learning\ rate$ 0.005 was chosen since it provided the best results terms of learning speed and obtained accuracy.

One of the most important parts of the LSTM network is the selection of variables $history\ size$ and $future\ target$. The larger $history\ size$ offers more historical data from which the network can learn more connections, and in many applications it is good to have a large $history\ size$. However, $history\ size$ of 24 was chosen for this model to cover the past 24 hours. This is sufficient due to the nature of the model that turns the heating on and off with a period of approximately 24 hours, and day_{ow} and day_w are added to the input to identify days in which the periodicity is disturbed (weekends and non-working days). Models with $history\ size$ greater than 24 do not bring any special progress, and reduce the learning speed, but also the accuracy of the model that in such situations incorrectly connects variables from the distant past with the future $T_{setpoint}$. For the $future\ target$, 24 hours was chosen to see how $T_{setpoint}$ would move on the horizon throughout the day, but a model with $future\ target$ 1 was also tested to compare with a rule based model that also does a 1 hour prediction.

IV. Relevant input variables selection procedure

This section compares learning with different sets of input variables for the considered four rooms and determines which set of the input variables gives the most accurate model.

To make it easier to determine the contribution of various variables to the $T_{setpoint}$ modelling accuracy, a correlation table is constructed. Figure 2 presents the correlation table for e.g. room 1.

Correlations are displayed only at a time when $T_{setpoint}$ is greater than 15, i.e. when the user has a control option. Correlation absolute values are taken into account because both positive and negative correlations are valuable for model learning. As expected, the largest absolute value of correlations with $T_{setpoint}$ have the variables w_{all} and w_{zone} generated using $T_{setpoint}$, and $system_{on/off}$ is not shown in the Fig. 2 because its value in these situations is always 1. Among other inputs, the highest correlation with $T_{setpoint}$ has T_{out}, then rH (the same correlation value for operating hours has $rH_{improved}$), then in order: I_g, I_{diff}, day_w, etc. It is interesting to note that T_{zone} has the lowest correlation, although at fast inference it seems like a variable that should have a high correlation with $T_{setpoint}$.

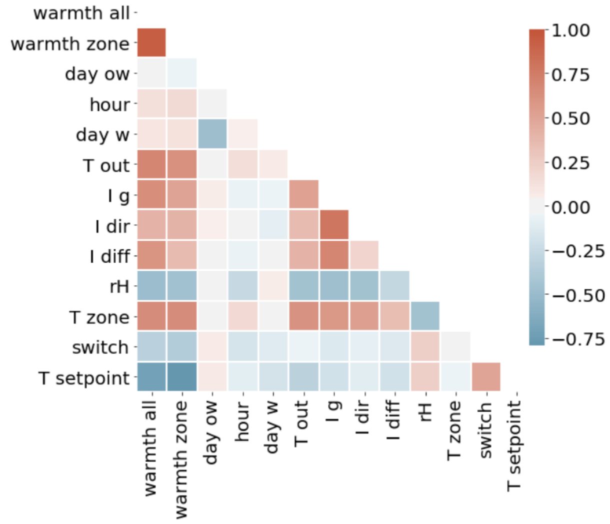

Fig. 2. Correlations between all available variables in e.g. room 1

Using the correlations from Fig. 2, variables that are highly correlated with predicted $T_{setpoint}$ and T_{zone} are more commonly used in multiple models. Models obtained with different sets of variables were tested. Table III shows the average model error for training, validation, and test depending on the set of input variables the model uses. Errors refer to models that predict the next 24 hours. Each set of variables was tested twice for each room and a better model was selected. The error for one room is calculated by taking the errors for all $T_{setpoint}$ predictions expressed in °C and calculating the mean square error. The table lists the average errors calculated as the arithmetic mean of the errors of all 4 considered rooms.

First, all available input variables are chosen. The results of this test are shown in row 1. In row 2, an LSTM with only 2 most relevant inputs is tested: $T_{setpoint}$ from the last 24 hours, which is usually similar to future $T_{setpoint}$, and day_{ow} by which the network can decide whether to follow last $T_{setpoint}$ or set a lower temperature, e.g. on Saturday afternoon or Sunday. Average error is about 35% less than in the first row due to the excessive total number of entries in the first row (24×11). Output $T_{setpoint}$ has a relatively simple trend, so with too many variables it is more prone to overfitting. These two variables with the addition of some more were used in all rows. The goal was to cover everything the network can learn from user behavior with as little unnecessary information as possible.

In rows 3, 4 and 5, $switch$ and $system_{on/off}$ and $hour$ are added, respectively. All three variables are intended to allow the network to better assess when the system is on and when it is not. In practice, the $switch$ proved to be a poor choice in rooms where the manual mode was often used. The other two variables gave similar results, and $system_{on/off}$ was chosen for slightly better accuracy and for variable simplicity.

Next, in rows 6, 7 and 8, the variables T_{zone}, T_{out} and I_g were tested. It is to be expected that the user most often

TABLE III
MODEL ACCURACY COMPARISON ACCORDING TO DIFFERENT SETS OF INPUT VARIABLES

	Input variables	Mean squared error (All 4 rooms average)		
		train	val.	test
1	day_{ow}, $hour$, day_{w}, T_{out}, I_{dir}, I_{diff}, I_{g}, rH, T_{zone}, $switch$, T_{setpoint}	0.485	0.765	0.940
2	day_{ow}, T_{setpoint}	0.418	0.510	0.625
3	day_{ow}, T_{setpoint}, $switch$	0.459	0.533	0.675
4	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$	0.429	0.477	0.566
5	day_{ow}, T_{setpoint}, $hour$	0.469	0.522	0.573
6	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, T_{zone}	0.450	0.540	0.607
7	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, T_{out}	0.418	0.565	0.631
8	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, I_{g}	0.414	0.518	0.640
9	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, rH	0.456	0.539	0.619
10	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, w_{zone}	0.421	0.472	0.557
11	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, w_{all}	0.438	0.491	0.548
12	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, rH_{improved}	0.372	0.481	0.545
13	day_{ow}, T_{setpoint}, $system_{\text{on/off}}$, rH_{improved}, w_{zone}	0.444	0.530	0.567

changes $T_{setpoint}$ when it is hot or cold, which is affected by indoor and outdoor temperature, but also solar irradiance if the room is lit by the sun during the day (rooms 1, 3 and 4 are oriented south, 2 north). However, the data shows that many users are not used to changing $T_{setpoint}$ too often, so they leave the temperature that is good enough, but not ideal. Also, most $T_{setpoint}$ changes have slightly different causes as described below. Additionally, these temperatures change even when the heating system is turned off, which is why all 3 variables reduce the accuracy of the model. However, some of these variables carry important information and can be used with customization.

Row 9 shows the testing of adding the variable rH, which does not increase the accuracy of the model compared to row 6, but the error is lower than in rows 7 and 8. Based on lower error and higher correlation rH with $T_{setpoint}$, a new variable $rH_{improved}$ was added in row 12.

The problem of infrequently changing $T_{setpoint}$ cannot be respected programmatically, but the problem of redundant information has is taken into account. Therefore, rows 10 and 11 introduce the variables w_{zone} and w_{all}, which, in addition to ignoring the parts where the system is off, also use current $T_{setpoint}$ to make it easier to infer when the user feels too hot. The coefficients $a = 1$, $b = 0.05$ and $c = 0.01$ (equation 5) were chosen for w_{all} because the combination of variables T_{zone}, T_{out} and I_{diff} with these 3 parameters gave the best results in testing, i.e. it best described the user's feeling of the temperature. Variable w_{zone} reduces the average error by 2% and w_{all} by 3% compared to row 4.

In row 12, the addition of a variable $rH_{improved}$ showing the amount of humidity in the air was tested while ignoring the relative humidity value when the system is off. Its addition to the input of the network from row 4 brought a more significant improvement than the variables from rows 10 and 11 (by about 4%). Model from this row proved to be the most accurate and Fig. 3 shows comparison of predicted and realized future plots for this model in room 1 on two different timestamps for which the prediction is made.

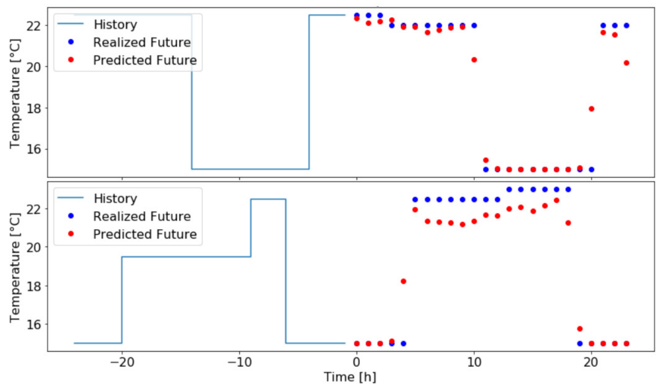

Fig. 3. Predicted and realized future plots for room 1

Combining $rH_{improved}$ with the w_{zone} and w_{all} variables in various ways has also been tested (one way is in row 13), but the best results are obtained by using only $rH_{improved}$ without data on T_{zone}, T_{out}, I_g, I_{dir} and I_{diff}. Although the overall error in this case is higher, it should be noted that the model in the room 1, which has the largest changes in $T_{setpoint}$, showed lower mean squared error, while for the other three rooms the error was higher. It may be because there were too few $T_{setpoint}$ changes for the model with two variables that measure the state of the rooms 2, 3 and 4 and the environment or because user in room 1 is more affected by temperature.

From this testing, it can be concluded that the relative humidity variable contains most of the relevant information not caused by the behavior of the algorithm that has so far controlled $T_{setpoint}$ most of the time, until the user intervenes. Relative humidity contains data on solar irradiance and outdoor temperature by which the behavior was attempted to be modeled in row 10. Higher solar irradiance and higher air temperature in an area are mainly associated with lower humidity. Also, a higher amount of humidity in the air leads to saturation of the air with water vapor, which then causes fog, dew, clouds and precipitation, which causes people to experience lower temperatures than they really are, and sometimes they want to warm up more than usual. Precipitation can often reduce the frequency of window opening, which can cause the room to overheat after the heating has been running for several hours, and users in such conditions more often resort to lowering $T_{setpoint}$ in the afternoon. Detection of such situations was the main task of the variables w_{zone} and w_{all}, but it turned out that $rH_{improved}$ is better in recognizing such situations, probably because it makes it easier to distinguish when will user reduce heating, and when he will open the window. This proves the importance of the psychological factor of the users. Although it does not have a large impact on air quality, humidity has a large impact on the air quality experience [15]. Lower humidity is also associated with pleasant weather, which is why people tend to spend more time outside and the warmth of home is less interesting than when the weather is worse.

V. Results

After selecting the ideal set of inputs, the LSTM model was compared with the model currently in use in the listed rooms. Rule based model generates temperature setpoint predictions. The initial assumption on which the rule based model is based is that people rarely change $T_{setpoint}$, which would make the application of machine learning methods. The aim here is was to examine whether LSTM can surpass the rule based model in $T_{setpoint}$ prediction. For the purposes of this paper, a rule based model called the "1h algorithm" was created. The "1h algorithm" maps $T_{setpoint}$ value from the last hour to the next from 06:00 to 18:00 during workdays, and from 06:00 to 15:00 on Saturdays. In the evening and on Sundays the system is usually turned off and the setpoint is set to 15 °C.

Also, in order to better benchmark the predictions of LSTM, a "24h algorithm" was created. The "24h algorithm" works by taking $T_{setpoint}$ values from the past 24 hours and persistently copying them to the next 24 hours in a single evaluation. Exceptionally, $T_{setpoint}$ from Saturday is used for Mondays, and on Sundays $T_{setpoint}$ is set to 15 °C, which brought the highest accuracy of such approach on the observed dataset.

Using these two algorithms, predictions were generated for the same datasets as for LSTM. Comparisons of the accuracy of the obtained LSTM model and the two algorithms for all four rooms are shown in Table IV. It may be observed that the LSTM model brings improvement of about 30%, measured by MSE on test data. Also, it may be observed that the LSTM has a higher error on the test data in room 4, while for other 3 rooms it reduces the error, especially in room 1. From this, it can be concluded that the algorithm has a higher effect in rooms where the average error is higher, i.e. in which users changed $T_{setpoint}$ temperature more often.

TABLE IV
Comparison of LSTM and rule based models MSE by rooms

Algorithm		**Room 1**	**Room 2**	**Room 3**	**Room 4**
LSTM 24h algorithm	train	0.681	0.232	0.329	0.244
	val	0.576	0.334	0.841	0.173
	test	1.072	0.283	0.649	0.174
rule based model 24h algorithm	train	3.857	0.913	0.974	0.956
	val	0.750	0.282	1.031	0.152
	test	1.873	0.316	0.615	0.090
LSTM 1h algorithm	train	0.353	0.103	0.159	0.098
	val	0.181	0.058	0.159	0.038
	test	0.310	0.049	0.142	0.034
rule based model 1h algorithm	train	1.498	0.787	0.864	0.564
	val	0.380	0.274	0.510	0.086
	test	0.431	0.143	0.309	0.038

VI. Conclusion

The aim of the paper is to predict the room temperature setpoint to improve the ambiance of the room and enable room air conditioning optimization. A three-layer LSTM neural network has been constructed to provide the time series prediction. The model predicts user defined room temperature setpoint 1 or 24 hours in advance with a resolution of one hour. Among the variables selected as inputs, are historical values for temperature setpoint, system on/off (system operation indicator), a modified relative humidity, and day of the week. This means that among the variables that need to be measured, it is sufficient to send only the relative humidity data to the input. For the temperature setpoint prediction during the next hour, an average MSE of 0.134 is obtained, which is 42% lower compared to 0.23 for the manual, rule based persistent model. If the relative humidity is not measured, it is possible to obtain approximately good results by measuring the temperature of the room.

VII. Acknowledgement

This work has been supported in part by Croatian Science Foundation under the project No. UIP-2020-02-9636 (project DECIDE - Distributed Control for Dynamic Energy Management of Complex Systems in Smart Cities).

References

[1] "Renewable Energy Policy Network for the 21st Century" (REN21). *Renewables 2020, Global Status Report, 2020.*

[2] A. Javed, H. Larijani, A. Ahmadinia, R. Emmanuel, M. Mannion, D. Gibson, "Design and Implementation of a Cloud Enabled Random Neural Network-Based Decentralized Smart Controller With Intelligent Sensor Nodes for HVAC", *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 393-403, 2017.

[3] J. Martinez-Gil, G. Chasparis, B. Freudenthaler, T. Natschlaeger, "Realistic user behavior modeling for energy saving in residential buildings", *25th International Workshop on Database and Expert Systems Applications*, pp. 68-72, 2014.

[4] N. I. Nwulu, "An artificial neural network model for predicting building heating and cooling loads", *2017 International Artificial Intelligence and Data Processing Symposium*, pp. 1-5, 2017.

[5] M. Li, Y. Li, X. Min, "Practice and Application of LSTM in Temperature Prediction of HVAC System" *IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020.

[6] S. Lee, L. F. Vecchietti, H. Jin, J. Hong, D. Har, "Power Management by LSTM Network for Nanogrids" in *IEEE Access*, vol. 8, pp. 24081-24097, 2020.

[7] Z. Afroz, T. Urmee, G. Shafiullah, G. Higgins, "Real-time prediction model for indoor temperature in a commercial building", *Applied Energy* vol. 231, pp. 29-53, 2018.

[8] C. Monna, D. Fay, K. N. Brown, N. Wilson, "Learning Occupancy in Single Person Offices with Mixtures of Multi-lag Markov Chains", *IEEE 25th International Conference on Tools with Artificial Intelligence* vol. 25, pp. 151-158, 2013.

[9] A. Martinčević, M. Vašak, V. Lešić, "Model Predictive Control for Energy-saving and Comfortable Temperature Control in Buildings", *Proc. of the 24th Mediterranean conference on control and automation*, pp. 298–303, 2016.

[10] M. Gulin, T. Pavlović, M. Vašak, "Photovoltaic panel and array static models for power production prediction: Integration of manufacturers' and on-line data", *Renewable Energy*, vol. 97, pp. 399-413, 2016.

[11] M. Gulin, T. Pavlović, M. Vašak, "A one-day-ahead photovoltaic array power production prediction with combined static and dynamic on-line correction", *Solar Energy*, vol. 142, pp. 49-60, 2017.

[12] S. Hochreiter, J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, pp. 1735-1780, 1997.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.

[14] D. P. Kingma, J. Ba, "Adam: a method for stohastic optimization", *International Conference on Learning Representations*, 2014.

[15] G. Clausen, P. O. Fanger, "Impact of Temperature and Humidity on the Perception of Indoor Air Quality", *Indoor Air*, vol. 8, pp. 80–90 1998.