e-ISSN: 3025-8189 ; p-ISSN: 3025-8197

LANCAH JURNAL INOVASI DAN TREN

doi.org/10.35870/ljit.v4i2.9838

Vol. 4, No. 2, 2026

PP. 2000-2018

Homepage http://www.journal.lembagakita.com/ljit

Multi-Room Time-Series Forecasting of Indoor Thermal Conditions Using Deep Learning Models

Muslimsyah[1*], Nizarli[1], Sofyan[1], Andri Novandri[2]

Department of Architecture and Planning, Universitas Syiah Kuala, Banda Aceh, Indonesia[1]

Department of Computer Engineering, Universitas Serambi Mekkah, Banda Aceh, Indonesia[2]

Corresponding Author's Email: muslimsyah@usk.ac.id

ARTICLE INFO

Article History:

Submitted	*11 12, 2026*
Accepted	*11 19, 2026*
Published	*11 26, 2026*

ABSTRACT

Temperature and humidity prediction are important aspects of indoor environmental condition monitoring. This study evaluates and compares three models: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Temporal Convolutional Network (TCN). The dataset consists of temperature and humidity measurements collected from four rooms, totaling eight observed variables. Dataset construction was performed using a sliding window approach with a lookback configuration of 6 steps and two forecasting horizons (2-step and 4-step), with a data sampling interval of 30 minutes. The experimental results indicate that model performance varies by variable type and forecasting horizon. For temperature prediction, the TCN model achieves the best performance at the 2-step forecasting horizon, with MAE of 0.281, RMSE of 0.344, and R^2 of 0.97. At the 4-step forecasting horizon, the GRU model outperforms the other models, achieving MAE of 0.403, RMSE of 0.519, and R^2 of 0.933. For humidity prediction, the GRU model demonstrates the best performance at both forecasting horizons, with MAE of 1.786 and R^2 of 0.897 at the 2-step forecasting horizon, and MAE of 2.014 and R^2 of 0.867 at the 4-step forecasting horizon. Furthermore, longer forecasting horizons lead to higher prediction errors. These findings indicate that the GRU model performs most consistently across the experiments, particularly for humidity prediction. In contrast, the TCN model shows promising potential for short-term temperature forecasting. This study highlights the potential of deep learning models to support multivariate data-driven indoor environmental monitoring and forecasting systems.

Keywords: *Multi-Room; Thermal; LSTM; GRU; TCN*

How to Cite:

Muslimsyah, M., Nizarli, N., Sofyan, S., & Novandri, A. (2026). Multi-Room Time-Series Forecasting of Indoor Thermal Conditions Using Deep Learning Models. LANCAH: Jurnal Inovasi Dan Tren, 4(2), 2000-2018. https://journal.lembagakita.com/ljit/article/view/9838

Copyright @ 2026. Published by Lembaga Komunitas Informasi Teknologi Aceh (KITA), Indonesia. This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Based on a work at http://journal.lembagakita.com/ljit

Abstract & Indexing

CC BY-NC 4.0

OPEN ACCESS

INTRODUCTION

Indoor temperature and humidity prediction are important aspects of monitoring and managing indoor environmental conditions. Temperature and humidity are two interrelated environmental parameters that can affect occupants' thermal comfort, indoor environmental quality, and the operational requirements of cooling and ventilation systems (Mustafaraj et al., 2011). Changes in these two parameters are influenced by various factors, such as outdoor environmental conditions, solar radiation intensity, the number of occupants, indoor activities, and the operation of electronic devices and air conditioning systems (Muslimsyah et al., 2024). Therefore, real-time monitoring alone cannot describe future environmental conditions. Forecasting temperature and humidity variations based on historical data can provide additional information to support decision-making and more effective indoor environmental management (Özbalta et al., 2012). In buildings with multiple rooms, temperature and humidity characteristics do not necessarily follow identical patterns of variation. Each room has different environmental conditions depending on building orientation, room area, ventilation, heat exposure, the number of occupants, and equipment usage (Muslimsyah et al., 2025). Furthermore, environmental changes in one room may exhibit temporal patterns that differ from those in other rooms. These conditions make predicting temperature and humidity across multiple rooms more complex than predicting within a single room (Mba et al., 2016). Prediction models must not only learn the temporal variations of individual variables but also process multiple variables collected from different rooms simultaneously (Hampo et al., 2026). Therefore, a multivariate time-series forecasting approach is highly relevant for representing indoor environmental dynamics more comprehensively (Attoue et al., 2018; Siregar & Wanto, 2017).

One of the main challenges in environmental time-series forecasting is enabling models to learn temporal patterns characterized by fluctuations. Conventional statistical methods can be used to model time-series patterns; however, their application may encounter limitations when dealing with data involving nonlinear relationships and multivariate temporal interactions. Advances in deep learning offer an alternative approach to capture these relationships through automatic feature extraction (Lv et al., 2022). Nevertheless, model selection remains a challenge because each approach has different learning mechanisms and characteristics. Forecasting accuracy may also vary as the prediction horizon increases. A model that performs well in short-term forecasting may not maintain the same performance when predicting several time steps (Khan & Maity, 2020). One widely used time-series forecasting model is Long Short-Term Memory (LSTM). LSTM extends Recurrent Neural Network (RNN) and captures long-term temporal dependencies through input, forget, and output gate mechanisms (Fang et al., 2021). In addition, the Gated Recurrent Unit (GRU) is an RNN architecture that uses update and reset gates to control information flow during temporal learning. GRU has a simpler structure than LSTM because it utilizes fewer gating mechanisms. This characteristic makes it an alternative for time-series modeling that requires learning temporal relationships with relatively lower architectural complexity (Chhetri et al., 2020; Li et al., 2026). In contrast, the Temporal Convolutional Network (TCN) uses causal and dilated convolutions to capture temporal patterns without relying on recurrent mechanisms. Causal convolution ensures that predictions at a given time step use only information from previous time steps. In contrast, dilated convolution expands the temporal receptive field that can be learned by the model (Liu et al., 2024; Zheng et al., 2025).

Although various studies have investigated indoor environmental prediction, several aspects still require further investigation. First, some studies focus on temperature prediction or a single specific zone, making the simultaneous evaluation of temperature and humidity across multiple rooms a relevant area for

further development. Second, the forecasting horizon can affect a model's prediction error. Therefore, evaluation should not be limited to a single prediction step but should also consider multiple future time steps. Based on these issues, a comparative study is needed to evaluate different models' ability to learn multivariate temporal relationships and predict temperature and humidity across multiple rooms. This study's main contribution is a comparative evaluation of LSTM, GRU, and TCN within a multivariate forecasting framework to simultaneously predict temperature and humidity across multiple rooms while considering the influence of different forecasting horizons. The findings are expected to clarify each model's strengths and limitations in learning indoor environmental temporal patterns and to support the development of more informative indoor environmental forecasting systems.

METHODS OF RESEARCH

Proposed Method

This study proposes a method for predicting indoor temperature and humidity across multiple rooms using three approaches: LSTM, GRU, and TCN. The proposed approach integrates temperature and humidity data collected from four rooms into a single multivariate dataset. Each room is represented by two observed variables, namely temperature and humidity, resulting in a dataset consisting of eight variables. This approach allows each model to learn the temporal patterns of temperature and humidity variations while capturing temporal information from all rooms simultaneously. The research process begins by collecting temperature and humidity data from four rooms. The data are then organized by measurement timestamp and transformed into a time-series dataset using a *sliding-window* approach. In this study, a 6-step *lookback* configuration with a 30-minute measurement interval is used, allowing the models to use three hours of historical data as input. Subsequently, the forecasting performance is evaluated using two forecast horizon configurations: a 2-step forecasting horizon, representing one hour ahead, and a 4-step forecasting horizon, representing two hours ahead. The dataset is chronologically divided into 70% training data, 15% validation data, and 15% testing data. During the modeling stage, all three models receive input data with six *lookback* steps and eight features and generate temperature and humidity predictions for all rooms over the specified forecasting horizon. The LSTM and GRU models are employed to learn temporal relationships through gating mechanisms in recurrent networks. In contrast, the TCN model utilizes causal convolution, dilated convolution, and residual connections to extract temporal patterns. Implementing these three models enables a comparative analysis of recurrent learning and temporal convolutional approaches for multivariate multi-step forecasting. The floor plan of the monitored rooms is presented in Figure 1, while the modeling process diagram is illustrated in Figure 2.

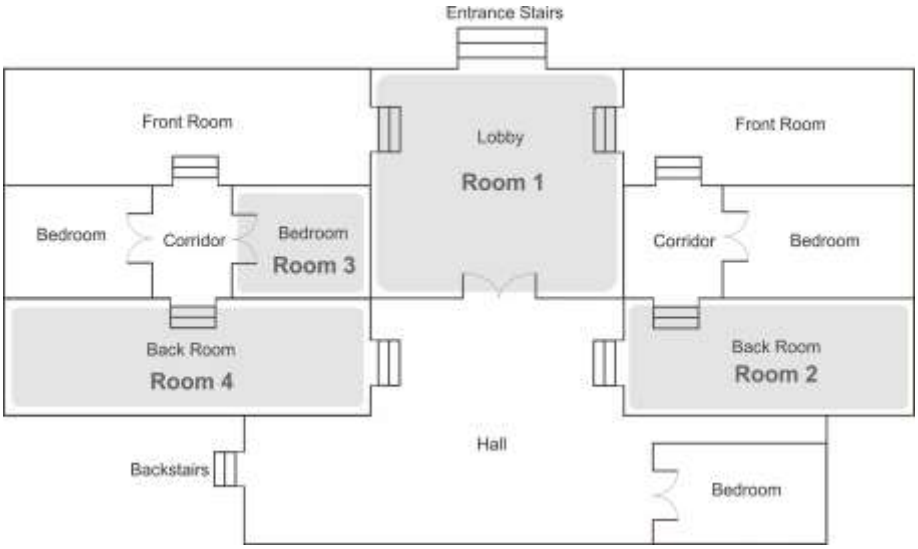

Figure 1. Location of the Rooms Being Tested
Source: Authors' documentation (2026)

In the first stage, temperature and humidity data are collected from each room. Each room has two measurement parameters: temperature (T_i) and humidity (H_i). Thus, each room produces two observed variables, for a total of eight variables across the dataset ($F = 8$). In the second stage, we construct a time-series dataset from the collected measurement data. At this stage, all variables from the four rooms are combined into a single feature matrix consisting of eight variables. Next, the dataset is structured using a *sliding-window* approach with two main parameters: *lookback* (L) and forecast horizon (H). The *lookback* parameter represents the number of historical observations used as model inputs, whereas the forecast horizon indicates the number of future time steps to be predicted. In this study, a *lookback* $L = 6$ is employed, meaning that the model uses the six previous observations as input data. With this configuration, each input sample has dimensions $L \times F$, while the prediction target has dimensions $H \times F$. The third stage involves developing and comparing three models: LSTM, GRU, and TCN. The LSTM model consists of two LSTM layers with 64 and 32 neurons, respectively. A *dropout* rate of 0.2 is applied between the layers to reduce the risk of *overfitting*. The temporal features extracted by the LSTM layers are subsequently passed to a *Dense* layer containing 16 neurons before generating the predictions. The GRU model employs a similar architecture but replaces the LSTM layers with GRU layers. Meanwhile, the TCN model adopts a temporal convolution approach through several *residual blocks*. Each block consists of *Conv1D* layers with causal convolution, *spatial dropout*, *layer normalization*, and *residual connections*. In this study, several *residual blocks* with different dilation rates are employed, ranging from $d = 1$ to $d = 4$. After temporal feature extraction, the resulting feature representations pass through a *Dense* layer with 16 neurons to generate temperature and humidity predictions for the specified forecasting horizon.

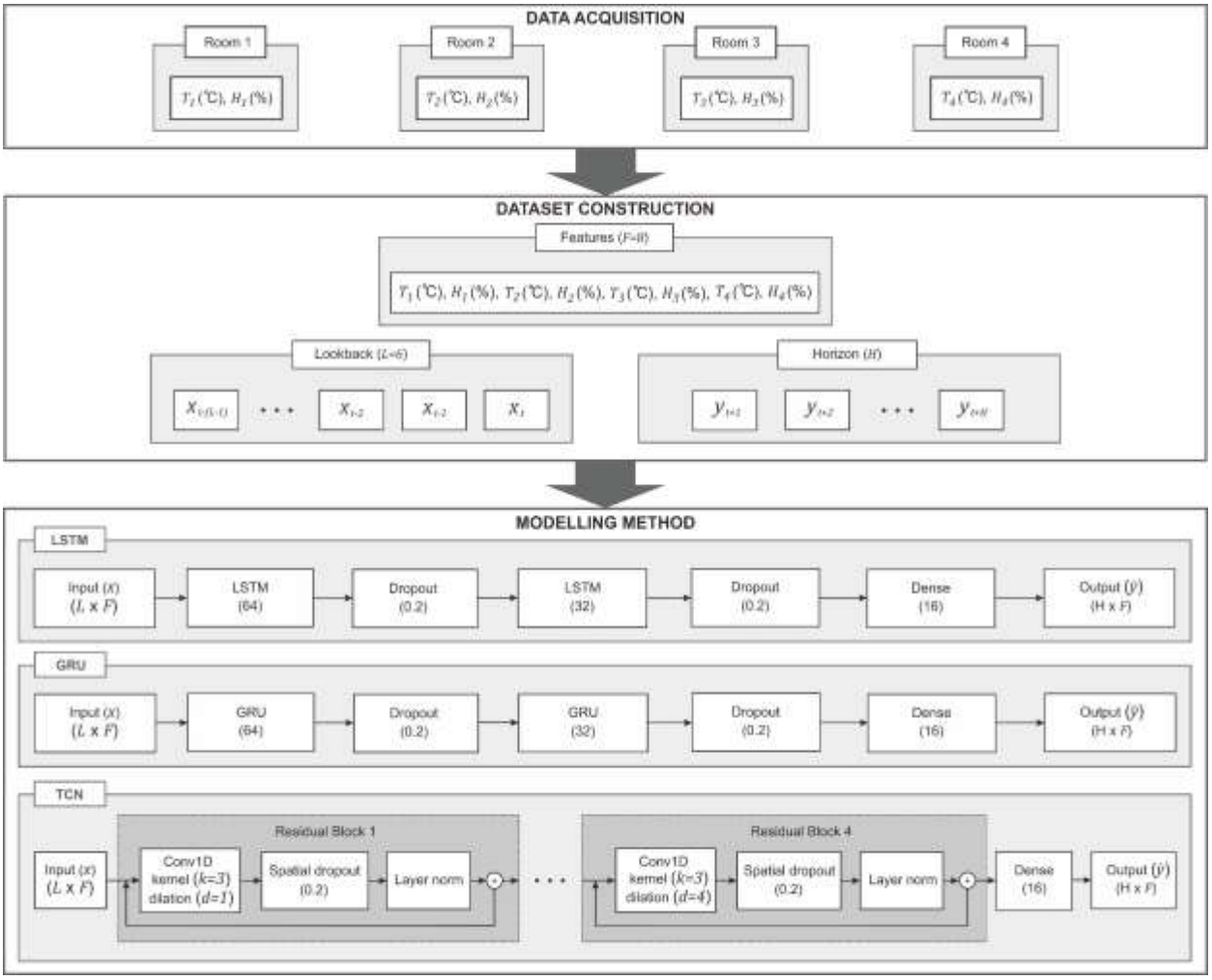

Figure 2. Diagram of the Proposed Method
Source: Authors' design (2026)

Long Short-Term Memory Model

LSTM is a type of RNN architecture designed to learn temporal dependencies in time-series data by utilizing a memory cell mechanism. This mechanism enables the model to retain important information from previous observations and mitigate the *vanishing gradient* problem commonly encountered in conventional RNN. In this study, the LSTM model receives multivariate data consisting of six *lookback* time steps and ten input variables, representing temperature and humidity measurements from five rooms. The temporal information learning process is performed through three main gating mechanisms: input gate, forget gate, and output gate (Fredyan & Kusuma, 2022). The input gate determines how much new information from the input data is stored in the LSTM memory cell. Mathematically, the input gate is defined as follows.

$$\boldsymbol{i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)} \quad (1)$$

where, x_t represents the current input data, h_{t-1} denotes the hidden state from the previous time step, W_i is the weight matrix connecting the current input data to the input gate, U_i is the weight matrix connecting

the previous hidden state to the input gate, b_i is the bias parameter added to the computation to help regulate the activation of the input gate, and σ denotes the sigmoid activation function. Furthermore, the forget gate determines which information from the previous memory to retain or discard. Mathematically, the forget gate is defined as follows.

$$\boldsymbol{f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right)} \tag{2}$$

This gate uses the current input data and the previous hidden state to generate an activation value. To construct the candidate new information, the candidate cell state is calculated using the following equation.

$$\boldsymbol{\widetilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C)} \tag{3}$$

This equation's result is called the candidate cell state because it is not stored in memory immediately. Subsequently, the cell state is updated by combining two components: the retained information from the previous memory and the new information to be stored. Mathematically, the cell state update is defined as follows.

$$\boldsymbol{C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t} \tag{4}$$

where, $f_t \odot C_{t-1}$ represents the previous information retained based on the forget gate value, while $i_t \odot \tilde{C}_t$ represents the new information incorporated based on the input gate value. These two components are added together to produce the updated memory, C_t. The output gate then determines which part of the cell state is used to generate the output at the current time step. Mathematically, the output gate is defined as follows.

$$\boldsymbol{o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)} \tag{5}$$

Based on the updated cell state and the output gate value, a new hidden state is generated using the following equation.

$$\boldsymbol{h_t = o_t \odot \tanh(C_t)} \tag{6}$$

where, h_t represents the information propagated to the next time step or the subsequent network layer. The LSTM model consists of two recurrent layers. The first layer generates a sequence of hidden states that is passed to the second layer, with dropout applied to reduce the risk of overfitting. The temporal representation extracted by the second layer is then passed to a fully connected layer with a ReLU activation function. Finally, the output is mapped to ten output variables for each prediction step. This mapping process can be expressed as follows.

$$\boldsymbol{\hat{y} = \mathrm{Reshape}\left(W_y \mathrm{ReLU}(W_d h + b_d) + b_y\right)} \tag{7}$$

where, $\hat{y}$ represents the prediction matrix with dimensions $H \times F$. H denotes the forecasting horizon and F represents the features corresponding to the temperature and humidity variables across all rooms.

Gated Recurrent Unit Model

GRU extends the RNN architecture by adding an information-control mechanism through two main gates: update gate and reset gate. The update gate determines how much information from the previous hidden state should be retained when generating a new hidden state (Yousaf et al., 2026). Mathematically, the update gate is defined as follows.

$$\boldsymbol{z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)} \tag{8}$$

The reset gate controls how much historical information is used to generate a candidate new hidden state. Mathematically, the reset gate is defined as follows.

$$\boldsymbol{r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)} \tag{9}$$

Subsequently, the relevant historical information required to construct a new representation is determined using the candidate hidden state equation, which is expressed as follows.

$$\boldsymbol{\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)} \tag{10}$$

The hidden state is then updated by combining historical information and candidate new information based on the update gate value. This combination process is performed using the hidden state update equation, which is expressed as follows.

$$\boldsymbol{h_t = z_t \odot h_{t-1} + (1 + z_t) \odot \tilde{h}_t} \tag{11}$$

The GRU model employed in this study consists of two recurrent layers. The first layer generates a sequence of temporal representations that is passed to the second layer, with dropout applied as a regularization mechanism. Subsequently, the final representation from the second layer is processed through a fully connected layer with a ReLU activation function. The output layer uses a linear activation function to generate multivariate predictions over multiple future time steps (Zhou et al., 2021). Mathematically, the GRU output mapping process can be expressed as follows.

$$\boldsymbol{\hat{y} = \text{Reshape}\big(W_y \text{ReLU}(W_d h_T + b_d) + b_y\big)} \tag{12}$$

where, h_T represents the hidden state at the final time step of the second GRU layer. As in the LSTM model, the model output has dimensions $H \times F$.

Temporal Convolutional Network Model

TCN is a convolution-based ANN architecture designed to learn temporal patterns in time-series data. TCN uses causal dilated convolutions to extract temporal information. Causal dilated convolution ensures that predictions at a given time step depend only on information from the current and previous time

steps, preventing the use of future information during learning (Nuo et al., 2025). The causal dilated convolution equation is expressed as follows.

$$\boldsymbol{F(t) = \sum_{i=0}^{k-1} W_i x_{t-di} + b} \tag{13}$$

where, $F(t)$ represents the convolution output, k denotes the kernel size, W_i represents the weights, x_{t-di} denotes the input data at a previous time step, d is the dilation rate that determines the spacing between data positions used in the convolution process, and b represents the bias. Subsequently, the input data are processed through the convolutional layer using the following equation.

$$\boldsymbol{u_t = \mathbf{LN}\left(\mathbf{Conv}_{k,d}(x)_t\right)} \tag{14}$$

Subsequently, a residual connection mechanism is applied to combine the convolution output with the original input through a shortcut pathway, denoted as $S(x)_t$. This process is expressed using the following equation.

$$\boldsymbol{u_t = \mathbf{LN}\left(\mathbf{Conv}_{k,d}(x)_t\right)} \tag{15}$$

Two residual blocks are arranged sequentially with different dilation rates. After the temporal feature extraction process, only the representation at the final time step is retained using a cropping operation. This representation is then flattened with a Flatten layer, passed through a fully connected layer with ReLU activation, and mapped to the output layer with a linear activation function (Wei et al., 2026). The TCN output generation process can be expressed as follows.

$$\boldsymbol{\hat{y} = \mathbf{Reshape}\left(W_y \mathbf{ReLU}(W_d \mathbf{Flatten}(Z_T) + b_d) + b_y\right)} \tag{16}$$

where, Z_T represents the temporal feature representation at the final time step after passing through two residual blocks, the final output consists of multivariate predictions with dimensions $H \times F$.

RESULTS

Model Performance Based on Temperature

Table 1 presents the temperature prediction performance of the LSTM, GRU, and TCN models across four rooms and two forecasting horizons. At the 2-step forecasting horizon, GRU achieves the lowest MAE, MSE, and RMSE values in Room 1 and Room 4, while TCN obtains the best results across all evaluation metrics in Room 3. In Room 2, TCN achieves the lowest MSE and RMSE, whereas LSTM produces the lowest MAE. At the 4-step forecasting horizon, the best-performing model varies across rooms. TCN achieves the lowest MAE in Room 1, LSTM performs best in Room 2, while GRU obtains the lowest MAE in Room 3 and Room 4. These results indicate that no single model consistently achieves the lowest prediction error across all rooms and forecasting horizons. Overall, GRU demonstrates relatively consistent performance, particularly at the 4-step forecasting horizon, where it achieves the lowest MAE in two of the

four rooms and performs well across several evaluation metrics. In contrast, TCN shows strong performance at the 2-step forecasting horizon, particularly in Room 3, where it achieves the best results across all evaluation metrics. LSTM also remains competitive, particularly in Room 2 at the 2-step forecasting horizon and in Room 2 at the 4-step forecasting horizon. These findings suggest that model performance is influenced by both the selected architecture and the temporal characteristics of each room. Furthermore, the comparison between the 2-step and 4-step forecasting horizons generally shows an increase in prediction errors at the longer forecasting horizon. This indicates that predicting indoor temperature further into the future presents greater challenges for the evaluated models. However, the magnitude of performance degradation varies across models and rooms, highlighting the importance of evaluating each room individually.

Table 1. Performance Results of Each Model for Temperature Prediction

Horizon Step	Model	Metric	Room 1	Room 2	Room 3	Room 4
2	LSTM	MAE	0.34	0.299	0.356	0.305
		MSE	0.159	0.172	0.176	0.153
		RMSE	0.399	0.415	0.42	0.391
		R^2	0.956	0.965	0.948	0.96
	GRU	MAE	0.231	0.304	0.306	0.236
		MSE	0.082	0.177	0.137	0.108
		RMSE	0.287	0.421	0.37	0.328
		R^2	0.977	0.963	0.96	0.972
	TCN	MAE	0.275	0.335	0.236	0.276
		MSE	0.126	0.165	0.078	0.114
		RMSE	0.355	0.406	0.28	0.337
		R^2	0.965	0.966	0.977	0.971
4	LSTM	MAE	0.401	0.435	0.418	0.481
		MSE	0.217	0.369	0.243	0.368
		RMSE	0.466	0.607	0.493	0.607
		R^2	0.942	0.926	0.931	0.907
	GRU	MAE	0.395	0.446	0.367	0.405
		MSE	0.242	0.335	0.211	0.298
		RMSE	0.491	0.578	0.459	0.546
		R^2	0.935	0.933	0.94	0.925
	TCN	MAE	0.373	0.475	0.505	0.416
		MSE	0.205	0.389	0.346	0.27
		RMSE	0.453	0.624	0.588	0.52

	R^2	0.945	0.922	0.901	0.932

Source: Authors' experimental results (2026)

Model Performance Based on Humidity

Table 2 presents the humidity prediction performance of the three deep learning models across the four rooms. At both the 2-step and 4-step forecasting horizons, GRU achieves the lowest MAE, MSE, and RMSE values and the highest R^2 values in all four rooms. The GRU model obtains MAE values ranging from 1.544 to 1.974 at the 2-step forecasting horizon and from 1.817 to 2.404 at the 4-step forecasting horizon. Meanwhile, TCN produces higher prediction errors than LSTM and GRU in most cases, particularly at the 4-step forecasting horizon. The evaluation results show that the performance of each model differs depending on the predicted variable and forecasting horizon. The evaluation results indicate that GRU is the most consistent model for humidity prediction across all four rooms, for both forecasting horizons 2 and 4. The GRU model's superiority is reflected in its lower MAE, MSE, and RMSE, as well as higher R^2 values. Meanwhile, the LSTM model remains competitive, particularly at horizon 4. In contrast, the TCN model exhibits limitations in predicting humidity, especially at longer forecasting horizons. These findings suggest that the GRU model is more suitable for the humidity data characteristics considered in this experiment, particularly when the system is required to perform simultaneous multivariate predictions across multiple rooms.

Table 2. Performance Results of Each Model for Humidity Prediction

Horizon Step	Model	Metric	Room 1	Room 2	Room 3	Room 4
2	LSTM	MAE	2.143	1.582	1.805	2.237
		MSE	6.543	3.744	4.723	7.118
		RMSE	2.558	1.935	2.173	2.668
		R^2	0.883	0.93	0.856	0.84
	GRU	MAE	1.95	1.544	1.677	1.974
		MSE	5.368	3.58	3.865	5.828
		RMSE	2.317	1.892	1.966	2.414
		R^2	0.904	0.933	0.882	0.869
	TCN	MAE	2.189	1.696	1.684	2.159
		MSE	7.031	4.584	4.085	6.156
		RMSE	2.652	2.141	2.021	2.481
		R^2	0.874	0.914	0.875	0.861
4	LSTM	MAE	2.561	2.069	2.013	2.398
		MSE	9.478	7.688	5.975	9.345
		RMSE	3.079	2.773	2.444	3.057
		R^2	0.832	0.857	0.82	0.794
	GRU	MAE	2.404	1.817	1.821	2.013
		MSE	8.221	5.177	4.793	6.693

		RMSE	2.867	2.275	2.189	2.587
		R^2	0.854	0.904	0.856	0.852
	TCN	MAE	2.923	3.078	2.542	3.067
		MSE	11.736	14.292	8.781	12.955
		RMSE	3.426	3.78	2.963	3.599
		R^2	0.792	0.735	0.736	0.714

Source: Authors' experimental results (2026)

Actual vs Predicted Results

This section presents a graphical comparison between actual temperature measurements and predictions from three deep learning models: LSTM, GRU, and TCN. In Figures 3 and 4, the LSTM and GRU models show relatively good ability to track the variation patterns of actual temperature, particularly during the temperature increase from morning to midday. Both models represent the rising temperature trend until it reaches its peak and then follow the decreasing trend from afternoon to evening. Meanwhile, the TCN model exhibits several deviations during the initial observation period, particularly when the temperature remains relatively stable. In Room 1, TCN tends to predict lower values than the actual measurements at several morning observation points. A similar pattern also appears in Room 2, although TCN tracks temperature variations as the increasing phase begins. These differences indicate that each model responds differently when learning temperature fluctuations under relatively stable environmental conditions and during periods of temperature variation. In Figures 5 and 6, all three models capture the increasing and decreasing temperature patterns; however, the differences are more noticeable around the peak temperature period. In Room 3, LSTM and GRU generate predictions relatively close to the actual temperature during the increasing phase, whereas TCN deviates around 15:00, when its predictions exceed the actual temperature. Following this period, all three models tend to follow the decreasing temperature trend, although differences in predicted values remain at several observation points toward the evening. In Room 4, LSTM and GRU show prediction patterns relatively close to the actual temperature, particularly during the increasing phase and around the peak temperature period. In contrast, TCN tends to predict lower temperatures during the peak period. These observations suggest that each model's ability to represent peak temperature variations may differ across rooms.

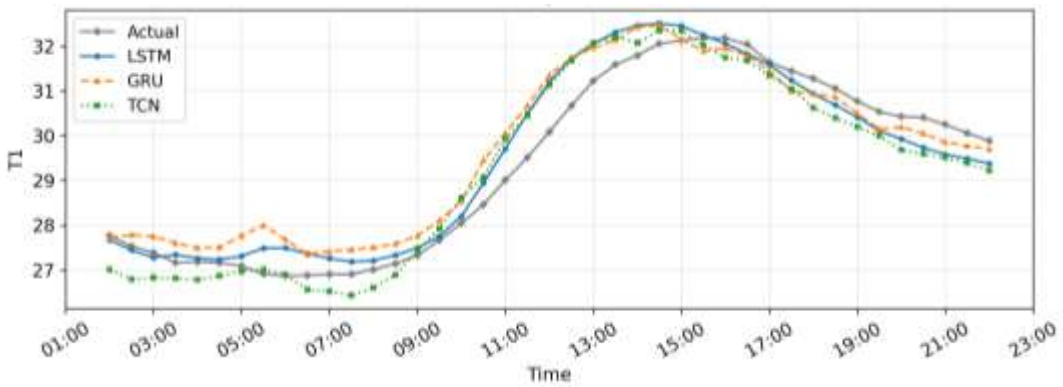

Figure 3. The Comparison of Temperature Graph on Test Results in Room 1
Source: Authors' experimental results (2026)

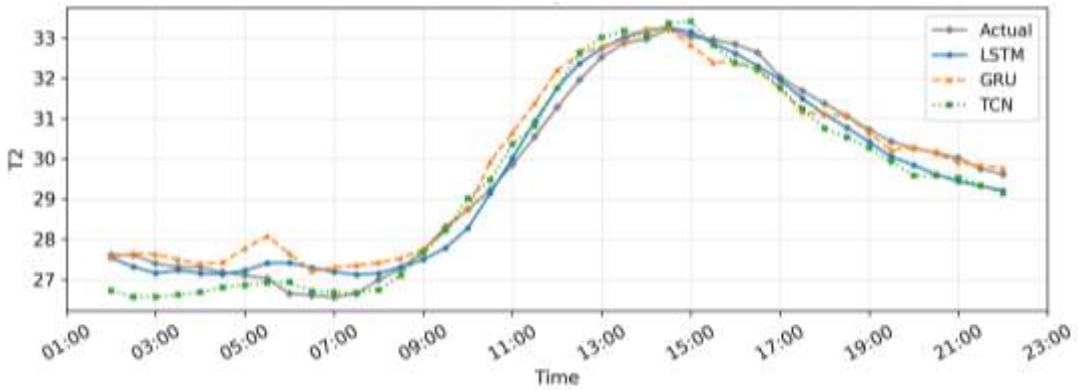

Figure 4. The Comparison of Temperature Graph on Test Results in Room 2
Source: Authors' experimental results (2026)

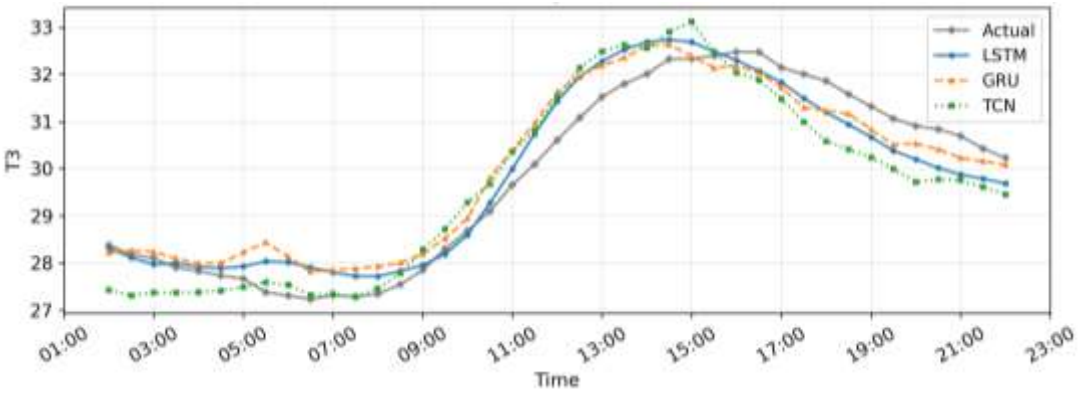

Figure 5. The Comparison of Temperature Graph on Test Results in Room 3
Source: Authors' experimental results (2026)

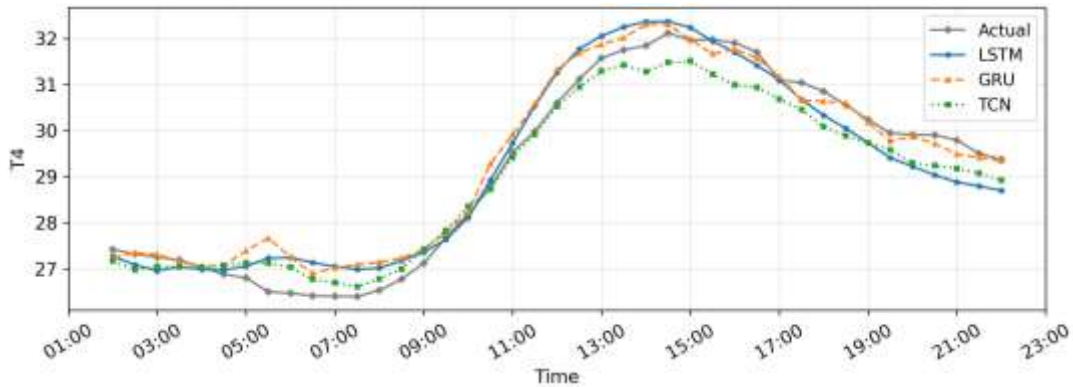

Figure 6. The Comparison of Temperature Graph on Test Results in Room 4
Source: Authors' experimental results (2026)

In Figure 7, the actual humidity values are relatively high at the beginning of the observation period, then gradually decrease to a minimum. Differences can be observed from the initial observation period

through the morning, during which the actual data exhibit several fluctuations that are not fully captured by the predicted results. The LSTM and GRU models show similar prediction patterns, whereas the TCN model tends to produce higher predicted values during certain periods, particularly toward the evening. In Figure 8, all three models represent the decreasing humidity pattern from morning to midday, followed by an increase from afternoon to evening. The LSTM and GRU models successfully capture the decrease in actual humidity values, approaching the minimum level, whereas the TCN model tends to produce higher predictions during this period. As humidity begins to increase, all three models again follow the actual trend relatively closely. This indicates that the LSTM and GRU models provide more appropriate responses in representing humidity variations during the period of significant decline in this room. In Figure 9, the actual humidity values also show a decreasing pattern from morning until reaching a minimum. The LSTM and GRU models capture this pattern relatively well, particularly during the decreasing and increasing humidity phases. Meanwhile, the TCN model tends to produce higher predictions than the actual data over most of the observation period, particularly in the early morning and evening. Although the TCN model captures the overall pattern of humidity variations, it still struggles to represent actual values during certain intervals accurately. In Figure 10, the prediction results of all three models show patterns similar to those observed in the other rooms: high humidity in the morning, a decrease toward midday, and a subsequent increase from afternoon to evening. The LSTM and GRU models capture the decreasing humidity pattern relatively well until it reaches its minimum. However, during the initial observation period, the actual data exhibit greater fluctuations than the predicted values. The TCN model shows a noticeable deviation after humidity reaches its minimum, producing predicted humidity values higher than the actual values. Nevertheless, all three models still capture the increasing humidity trend from afternoon to evening.

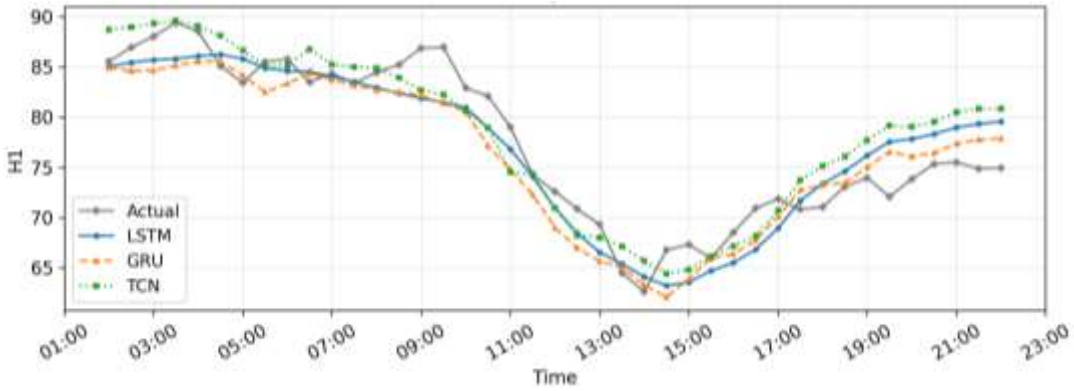

Figure 7. The Comparison of Humidity Graph on Test Results in Room 1
Source: Authors' experimental results (2026)

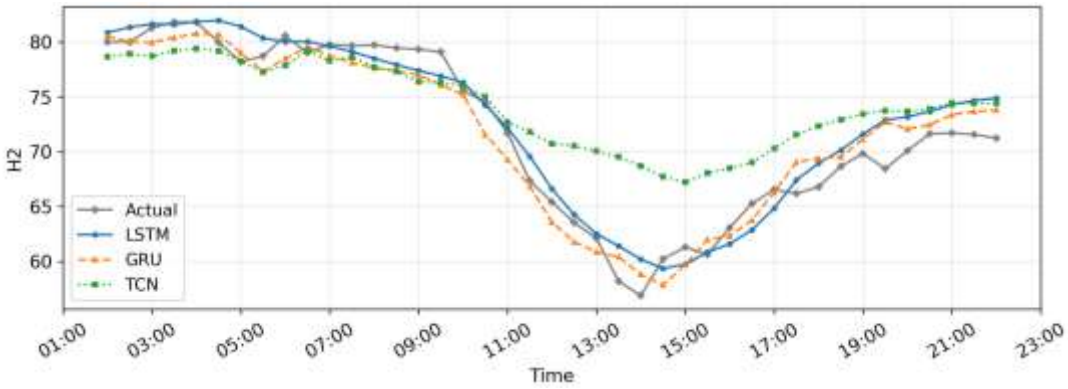

Figure 8. The Comparison of Humidity Graph on Test Results in Room 2
Source: Authors' experimental results (2026)

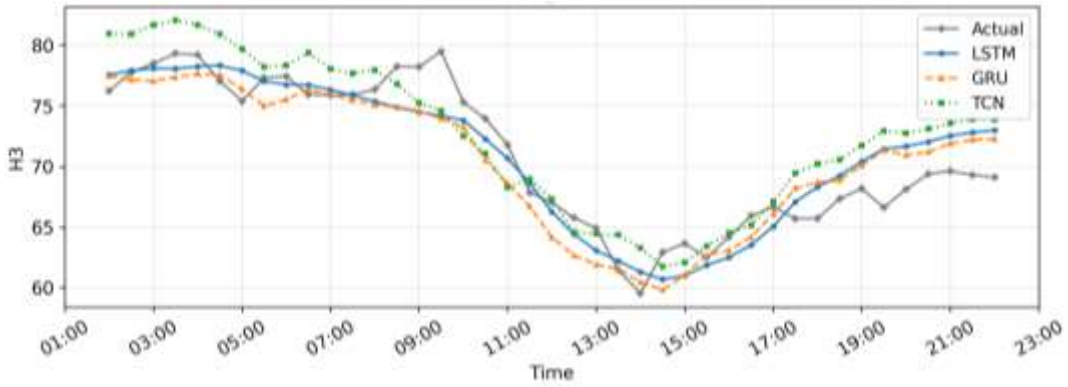

Figure 9. The Comparison of Humidity Graph on Test Results in Room 3
Source: Authors' experimental results (2026)

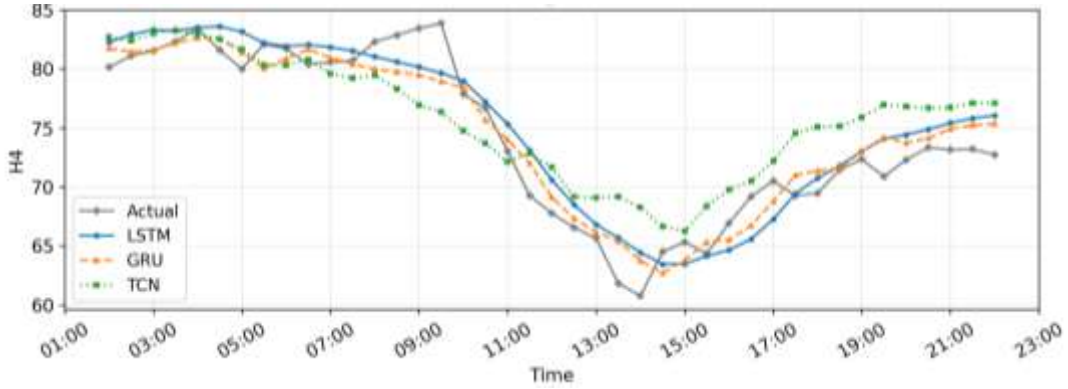

Figure 10. The Comparison of Humidity Graph on Test Results in Room 4
Source: Authors' experimental results (2026)

DISCUSSION

Effect of Forecasting Horizon on Prediction Performance

The forecasting horizon has a noticeable influence on the prediction performance of all three models. In general, increasing the forecasting horizon from 2-step to 4-step results in higher prediction errors and lower R^2 values. For example, the overall temperatures MAE of TCN increases from 0.301 at the 2-step forecasting horizon to 0.506 at the 4-step forecasting horizon. Similarly, the humidity MAE of GRU increases from 1.803 to 2.012 as the forecasting horizon becomes longer. This degradation in performance can be associated with the increasing uncertainty involved in predicting future environmental conditions from historical observations. As the prediction target moves further into the future, the relationship between the available input sequence and the target values becomes more challenging to learn. Consequently, models may have greater difficulty representing short-term fluctuations and changes in environmental patterns. These results emphasize the importance of considering the forecasting horizon when selecting a suitable model for indoor environmental prediction.

Differences in Temperature and Humidity Prediction

The comparison between temperature and humidity prediction reveals differences in how the evaluated models learn the temporal characteristics of each variable. Temperature prediction shows a more varied performance across the three architectures and four rooms. TCN performs well at the shorter forecasting horizon, whereas GRU provides better overall performance at the longer horizon. In contrast, humidity prediction demonstrates a more consistent pattern, with GRU achieving the best results across all rooms and both forecasting horizons. These differences suggest that temperature and humidity may exhibit distinct temporal characteristics within indoor environments. Although both variables are influenced by environmental conditions and indoor activities, their variations do not necessarily follow identical patterns. Therefore, a model that performs well for temperature prediction may not achieve the same level of accuracy for humidity prediction. The results support the need to evaluate forecasting models separately for different environmental variables, even when they are processed within the same multivariate forecasting framework.

Comparative Performance of Deep Learning Models

Based on the comparison in Table 3, the LSTM, GRU, and TCN models differ in their ability to predict temperature and humidity. At a forecasting horizon of 2-step, TCN achieves the lowest average MSE and RMSE values and the highest R^2 value for temperature prediction, with values of 0.121, 0.344, and 0.97, respectively. However, GRU obtains the lowest average MAE of 0.269. For humidity prediction at the same horizon, GRU outperforms the other models across all evaluation metrics, achieving MAE of 1.786, MSE of 4.66, RMSE of 2.147, and R^2 of 0.897. At a forecasting horizon of 4-step, GRU achieves the best average performance for both temperature and humidity prediction across all evaluation metrics. For temperature, GRU obtains MAE of 0.403, MSE of 0.272, RMSE of 0.519, and R^2 of 0.933. Similarly, for humidity, GRU achieves MAE of 2.014, MSE of 6.221, RMSE of 2.48, and R^2 of 0.867. These results indicate that GRU provides the most consistent overall performance, particularly at the longer forecasting horizon. Furthermore, increasing the forecasting horizon from 2-step to 4-step generally results in higher prediction errors and lower R^2 values across the evaluated models. This finding highlights the influence of forecasting horizon on prediction accuracy and suggests that model selection should consider both the

characteristics of the predicted variables and the required forecasting period.

Table 3. The Result of Comparison Models for Each Variable

Horizon Step	Variable	Model	$\overline{\text{MAE}}$	$\overline{\text{MSE}}$	$\overline{\text{RMSE}}$	$\overline{R^2}$
2	Temperature	LSTM	0.325	0.165	0.406	0.957
		GRU	0.269	0.126	0.351	0.968
		TCN	**0.281**	**0.121**	**0.344**	**0.97**
	Humidity	LSTM	1.942	5.532	2.334	0.877
		GRU	**1.786**	**4.66**	**2.147**	**0.897**
		TCN	1.932	5.464	2.324	0.881
4	Temperature	LSTM	0.434	0.299	0.543	0.926
		GRU	**0.403**	**0.272**	**0.519**	**0.933**
		TCN	0.442	0.303	0.546	0.925
	Humidity	LSTM	2.26	8.122	2.838	0.826
		GRU	**2.014**	**6.221**	**2.48**	**0.867**
		TCN	2.903	11.941	3.442	0.744

Source: Authors' calculation based on Tables 1 and 2 (2026)

CONCLUSION

Based on this study's findings, the LSTM, GRU, and TCN models can learn temporal patterns from multivariate data and predict multiple future time steps. All three models can represent temperature and humidity variations across the four rooms. However, their prediction accuracy varies by variable type, room characteristics, and forecasting horizon. The comparative results reveal that model performance depends on the prediction variable and forecasting horizon. For temperature forecasting, TCN performs best at the 2-step forecasting horizon, with MAE of 0.281, RMSE of 0.344, and R^2 of 0.970. At the 4-step forecasting horizon, GRU performs better, achieving MAE of 0.403, RMSE of 0.519, and R^2 of 0.933. For humidity forecasting, GRU consistently performs best at both horizons, with MAE of 1.786 and R^2 of 0.897 at the 2-step forecasting horizon and MAE of 2.014 and R^2 of 0.867 at the 4-step forecasting horizon. These findings indicate that no single deep learning architecture is universally superior for all forecasting configurations. TCN is more effective for short-term temperature prediction, whereas GRU demonstrates more consistent performance for longer-term temperature forecasting and humidity prediction. Furthermore, the results show that forecasting performance generally decreases as the prediction horizon increases, highlighting the added challenges of longer-term indoor environmental forecasting. Nevertheless, performance degradation varies across models and variables. The visualization results comparing actual and predicted data also show that all three models can capture the general patterns of temperature and humidity variations, including increasing and decreasing trends, as well as the occurrence of maximum and minimum values. However, deviations remain during certain observation periods, particularly when environmental conditions change significantly. Overall, GRU is the most consistent

model, particularly for humidity prediction and temperature prediction at longer forecasting horizons. Meanwhile, TCN demonstrates advantages in several short-horizon temperature prediction scenarios, whereas LSTM continues to deliver competitive results in several rooms. These findings confirm that no single model consistently outperforms the others across all variables and forecasting horizons. Therefore, model selection should consider the characteristics of the predicted variables and the required forecasting period.

AUTHOR CONTRIBUTIONS

The first author contributed to the conceptualization of the study, critical review, editing of the manuscript. The second author contributed to conceptualization of the study and critical review to the manuscript. The third author contributed to the development of the research methodology, critical review, editing of the manuscript. The fourth author contributed to the conceptualization of the study, interpretation of the research findings, development of the research methodology, data collection, data processing, data analysis, and preparation of the initial manuscript. All authors participated in the discussion of the research findings, reviewed and approved the final version of the manuscript, and are responsible for the integrity and accuracy of the work.

CONFLICT OF INTEREST

The authors declare that there are no financial or non-financial conflicts of interest that could have influenced the conduct of the research, data analysis, preparation, or publication of this article.

ARTIFICIAL INTELLIGENCE (AI) USE STATEMENT

In preparing this article, the authors used artificial intelligence (AI) technology as an assisting tool for language editing, grammar checking, and improving the structure and clarity of the manuscript. AI was not used to generate, manipulate, or analyze the research data. The authors retain full responsibility for the content of the article, interpretation of the findings, accuracy, and integrity of the manuscript.

REFERENCES

Attoue, N., Shahrour, I., & Younes, R. (2018). Smart building: Use of the artificial neural network approach for indoor temperature forecasting. *Energies, 11*(2), Article 395. https://doi.org/10.3390/en11020395

Chhetri, M., Kumar, S., Roy, P. P., & Kim, B. G. (2020). Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan. *Remote Sensing, 12*(19), Article 3174. https://doi.org/10.3390/rs12193174

Fang, Z., Crimier, N., Scanu, L., Midelet, A., Alyafi, A., & Delinchant, B. (2021). Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model. *Energy and Buildings, 245*, Article 111053. https://doi.org/10.1016/j.enbuild.2021.111053

Fredyan, R., & Kusuma, G. P. (2022). Spatiotemporal convolutional LSTM with attention mechanism for monthly rainfall prediction. *Communications in Mathematical Biology and Neuroscience, 2022*(1),

1–25. https://doi.org/10.28919/cmbn/7761

Hampo, C. C., Schinasi, L. H., & Hoque, S. (2026). Indoor temperature and humidity prediction in residential buildings: A review of the white box and black box modeling techniques. *Building and Environment, 290*, Article 114125. https://doi.org/10.1016/j.buildenv.2025.114125

Khan, M. I., & Maity, R. (2020). Hybrid deep learning approach for multi-step-ahead daily rainfall prediction using GCM simulations. *IEEE Access, 8*, 52774–52784. https://doi.org/10.1109/ACCESS.2020.2980977

Li, B., Wang, Z., Guo, Y., Zhou, X., Qiao, B., & Han, L. (2026). LSTM-GRU hybrid model for multi-layer microclimate prediction in solar greenhouse. *Scientific Reports, 16*(1), Article 18625. https://doi.org/10.1038/s41598-026-49691-w

Liu, S., Xu, T., Du, X., Zhang, Y., & Wu, J. (2024). A hybrid deep learning model based on parallel architecture TCN-LSTM with Savitzky-Golay filter for wind power prediction. *Energy Conversion and Management, 302*, Article 118122. https://doi.org/10.1016/j.enconman.2024.118122

Lv, R., Yuan, Z., Lei, B., Zheng, J., & Luo, X. (2022). Building thermal load prediction using deep learning method considering time-shifting correlation in feature variables. *Journal of Building Engineering, 61*, Article 105316. https://doi.org/10.1016/j.jobe.2022.105316

Mba, L., Meukam, P., & Kemajou, A. (2016). Application of artificial neural network for predicting hourly indoor air temperature and relative humidity in modern building in humid region. *Energy and Buildings, 121*, 32–42. https://doi.org/10.1016/j.enbuild.2016.03.046

Muslimsyah, M., Safwan, S., & Novandri, A. (2025). Comprehensive assessment of indoor thermal in vernacular building using machine learning model with GAN-based data imputation: A case of Aceh region, Indonesia. *Buildings, 15*(14), Article 2448. https://doi.org/10.3390/buildings15142448

Muslimsyah, Munir, A., Away, Y., Abdullah, Arif, T. Y., & Novandri, A. (2024). A model of indoor thermal condition based on traditional Acehnese houses using artificial neural network. *Heliyon, 10*(23), Article e40644. https://doi.org/10.1016/j.heliyon.2024.e40644

Mustafaraj, G., Lowry, G., & Chen, J. (2011). Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office. *Energy and Buildings, 43*(6), 1452–1460. https://doi.org/10.1016/j.enbuild.2011.02.007

Nuo, M., Zou, D., Liang, X., & Gao, D. (2025). Research on short-term energy consumption forecasting for cold regions based on the TCN–Transformer model. *Sustainability, 17*(22), Article 10230. https://doi.org/10.3390/su172210230

Özbalta, T. G., Sezer, A., & Yildiz, Y. (2012). Models for prediction of daily mean indoor temperature and relative humidity: Education building in Izmir, Turkey. *Indoor and Built Environment, 21*(6), 772–781. https://doi.org/10.1177/1420326X11422163

Siregar, S. P., & Wanto, A. (2017). Analysis of artificial neural network accuracy using backpropagation algorithm in predicting process (forecasting). *International Journal of Information System & Technology, 1*(1), 34–42. https://doi.org/10.30645/ijistech.v1i1.4

Wei, Z., Deng, Z., Dai, R., Zhang, H., & You, H. (2026). Operational functional-zone proxies as learnable context for TCN-based indoor temperature forecasting. *Buildings, 16*(15), Article 2966. https://doi.org/10.3390/buildings16152966

Yousaf, J., Harseno, R. W., & Yee, J.-J. (2026). Lightweight IoT-edge framework with attention-enhanced GRU for closed-loop real-time indoor environmental monitoring and prediction. *Measurement:*

Sensors, *44*, Article 101994. https://doi.org/10.1016/j.measen.2026.101994

Zheng, P., Zhang, W., Gao, B., Ma, Y., & Chen, C. (2025). Multi-step apparent temperature prediction in broiler houses using a hybrid SE-TCN–Transformer model with Kalman filtering. *Sensors*, *25*(19), Article 6124. https://doi.org/10.3390/s25196124

Zhou, C., Xu, Z., Wu, J., Liu, K., & Guan, X. (2021). Multi-zone indoor temperature prediction based on graph attention network and gated recurrent unit. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)* (pp. 1973–1978). IEEE. https://doi.org/10.1109/CASE49439.2021.9551630