Journal of Energy Storage 69 (2023) 107866

Contents lists available at ScienceDirect

Journal of Energy Storage

journal homepage: www.elsevier.com/locate/est

Research Papers

# Dynamic artificial neural network model for ultralow temperature prediction in hydrogen storage tank

Haeun Lee [a], Kiwan Kim [a], Daeyoung Kong [a], Min Hyeong Ahn [b], Donghoon Lee [c], Hobyung Jun [c], Chirag R. Kharangate [d], Jaiyoung Ryu [e], Hyoungsoon Lee [a,b,*]

[a] Department of Intelligent Energy and Industry, Chung-Ang University, Seoul 06974, South Korea
[b] School of Mechanical Engineering, Chung-Ang University, Seoul 06974, South Korea
[c] Institute of Gas Safety R&D, Korea Gas Safety Corporation, Yeongwol-gun, Gangwon-do 26203, South Korea
[d] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH 44106, USA
[e] Department of Mechanical Engineering, Korea University Seoul, Seoul 02841, South Korea

ARTICLE INFO

*Keywords:*
Deep-learning model
Dynamic artificial neural network
Recurrent neural network
Hydrogen storage
Compressed hydrogen

ABSTRACT

Hydrogen fuel cell vehicles are gaining popularity as an alternative to electric vehicles. To support their use, research on hydrogen-storage devices and charging stations is being conducted actively. However, high-pressure hydrogen dispensing can lead to a temperature increase and potential explosion hazards. Therefore, maintaining safe hydrogen temperatures during charging is crucial. However, accurately predicting temperature changes in hydrogen heat exchangers is challenging owing to rapidly changing operating conditions and large temperature differences. In this study, we adopt a deep-learning-based artificial neural network for predicting temporal temperature variations in low-temperature hydrogen heat exchangers. Two networks, namely, the deep neural network (DNN) and recurrent neural network (RNN), are constructed with Bayesian optimization based on the experimentally measured data from an ultralow-temperature hydrogen heat exchanger to predict temperature. Both models showed superior prediction accuracies of temperature within 3.11 % of mean absolute error (MAE) and heat transfer rates of the heat exchanger within 2.53 % of MAE. In particular, DNN modeling is recommended as a universal method for predicting the temperature when previous data are known, and RNN modeling is recommended for predicting the temperature for successive actions in a compressed-hydrogen system.

## 1. Introduction

The use of fossil-fuel-based energy has increased significantly because of rapid industrialization, leading to air pollution caused by the depletion and excessive use of fossil fuels. To address this problem, international environmental regulations are being strengthened to prevent global warming caused by carbon dioxide emissions. As part of this effort, there is a growing interest in eco-friendly renewable energy [1,2]. The transportation sector is actively replacing internal combustion engines in vehicles with alternative sources of energy to control carbon dioxide emissions. Hydrogen fuel cells represent a promising alternative energy source that can overcome the limitations of conventional battery electric vehicles (BEVs), such as long recharge times and short driving ranges [3]. Research is being conducted on various methods for storing hydrogen in fuel cell vehicles, including compressed gas, cryogenic storage, and metal hydrides [4]. Compressed gas storage requires hydrogen to be maintained at very high pressures (35–70 MPa) and low temperatures (−40 °C to −33 °C), which is typically achieved using heat exchangers. Accordingly, ensuring that the temperature remains within safe limits for rapid and secure hydrogen charging while avoiding hazardous reactions at elevated pressures and temperatures is crucial. However, controlling the temperature of hydrogen is challenging owing to abrupt changes caused by operating conditions such as flow rate, pressure, and ambient temperature. This can result in sudden changes in the heat loss to the surroundings. Therefore, most types of hydrogen equipment include oversized heat exchangers to actively control the hydrogen temperature. Thus, there is a clear need for high-accuracy modeling to predict the temporal thermal performances of hydrogen heat exchangers operating at low temperatures.

There are some reported studies on predicting the thermal performance of a hydrogen storage system using computational fluid dynamics simulations [5,6]. However, prediction through simulation has several limitations associated with the complex geometry and boundary

* Corresponding author at: Department of Intelligent Energy and Industry, Chung-Ang University, Seoul 06974, South Korea.
*E-mail address:* leeh@cau.ac.kr (H. Lee).

https://doi.org/10.1016/j.est.2023.107866
Received 3 February 2023; Received in revised form 4 May 2023; Accepted 27 May 2023

**Nomenclature**

| | |
|---|---|
| $c$ | cell state |
| $f$ | forget gate |
| $g$ | current gate |
| $h$ | output of hidden layer |
| $i$ | input gate |
| $\dot{m}$ | mass flow rate |
| $o$ | output gate |
| $P$ | pressure |
| $T$ | temperature |
| $x$ | data |
| $W$ | weight |
| $tanh$ | hyperbolic tangent function |

*Greek symbols*

| | |
|---|---|
| $\sigma$ | sigmoid function |

*Subscripts*

| | |
|---|---|
| *exp* | experimental |
| *in* | inlet |
| *out* | outlet |
| *pred* | prediction |
| *t* | current second |
| *t-1* | previous second |

conditions, which incur additional computational time and cost. Deep-learning-based artificial neural networks (ANNs) provide superior benefits as powerful tools for predicting the thermal and hydraulic performances of complex systems without the need for constructing the geometries and meshes of these systems. Moreover, they have been very successful for solving complex, nonlinear, dynamic, and multivariable problems and for tolerating errors, inaccuracies, and missing data [7]. The commonly used ANN architecture for thermal analyses is the deep neural network (DNN), which has been widely used over the past decade, especially for predictive modeling of renewable-energy thermal systems and hydraulic performance [8–14]. In related work, Islamoglu [9] predicted the heat transfer rate of a natural convection wire on a tube-type heat exchanger using DNN for modeling the condensers in the refrigeration system; the heat transfer rate was obtained based on 12 operational and geometrical input parameters and showed good predictive accuracies of 5.6 % and 7.9 % in terms of the mean absolute error (MAE) for training and testing, respectively. Lee et al. [10] and Kim et al. [11] also adopted DNN-based models for heat transfer analyses in micro-pin fin heat sinks; in their work, the friction factor and Nusselt number of the pin fin arrays were predicted using various operating conditions from other research sources. Their results indicated that the DNN model predicted the friction factor and Nusselt number with MAEs of 14.5 % and 7.5 %, respectively. In addition, modeling was developed for renewable energy such as PCM for thermal storage system [12], hybridization based on hydrogen storage [13], passive cooling systems in nuclear power plants [14], and energy production [15,16]. Recurrent neural network (RNN) is another type of ANN that is used for time-dependent learning. Xi et al. [17] used this approach to estimate the state of charge of lithium-ion batteries with errors less than 1 %.

Most previous DNN-based thermal predictive models were typically created for steady-state conditions at room temperature; thus, research on dynamic temperature prediction models has been rarely reported. In this study, we aim to predict the target temperature of hydrogen and the temporal thermal performance of a hydrogen heat exchanger during transient flow operation at low temperatures (−44 °C to 25 °C). To achieve this, we have utilized DNN and RNN based on time-series data. These models have been utilized to predict the variation in the target temperature of hydrogen over time in a compressed-hydrogen system.

## 2. Method

### *2.1. Experimental data from compressed hydrogen cooling facility*

We first experimentally measured the transient hydrogen temperatures at different ambient conditions with various mass flow rate changes over time. A prebuilt safety facility for leakage inspection of hydrogen vehicle components was modified to control the hydrogen to a relatively low temperature range of −25 to −50 °C, as shown in Fig. 1(a), while the simplified schematic used for obtaining the experimental data used in the neural network modeling is shown in Fig. 1(b). Hydrogen at room temperature is introduced into the facility from a 50 L high-pressed tank, and the mass flow is measured using a Coriolis flow meter (Oval, CA004G13). Thereafter, the temperature is adjusted to a low range of −25 to −50 °C as it passes through the solid-state heat exchanger of the cooling unit. This heat exchanger allows efficient cooling in the form of Alu cold fill (ACF) and optimally transfers the heat-flow-medium flowing into the evaporator to the coolant to quickly lower the temperature of hydrogen gas. Thereafter, the cooled hydrogen is transferred to the target location through a tube in the test facility (0.75 in. diameter, 5 m long). The temperature and pressure were measured at the inlet and outlet of the heat exchanger, target, and sample to determine the temporal changes caused by heat removal and natural convective heat addition in the heat exchanger.

The inlet and target hydrogen temperatures were measured using R221 thermocouples (WISE Vietnam, sheathed-type resistance) while the outlet temperature of the cooling unit and temperatures of the samples were measured using NTC thermistors [18] implemented in the cooling unit. The pressure was measured at the inlet of the cooing unit and the target. The detailed information and accuracies of the experimental measurements in this study are summarized in Table 1.

Table 2 shows the detailed test cases examined in this study. A total of six cases were tested according to the flow rate, environmental temperature conditions, and various test durations. In Cases 1 to 3, the experiments were conducted while maintaining a constant flow rate of 5–6 g/s with a relatively high ambient temperature of 19 °C. Case 4 was performed at a constant flow rate of 3–5 g/s and a lower ambient temperature of 12 °C. On the other hand, Cases 5 and 6 were conducted at variable flow rates of 0–80 g/s at ambient temperatures of 12 and 5 °C, respectively.

### *2.2. Modeling approaches*

#### *2.2.1. DNN and RNN*

The DNN is a feed-forward network composed of neurons, an input layer, a hidden layer, and an output layer; it enables nonlinear separation by combining two or more neurons. This architecture calculates the net input function value by combining the weight (W) set to a random value in each layer with the input data (X). After combining the weights and input data (WX), an appropriate bias (b) is added. Then, the activation function calculates the data (WX + b) to obtain the output value ($Y_{pred}$). The weight is changed in each layer so that the result and actual value ($Y_{exp}$) are within the allowable error ranges. The error value generated at the output layer is backpropagated to the hidden layer, and then the weights of the hidden layer are updated according to the error values generated in the hidden layer. The schematic representation of the DNN is shown in Fig. 2(a).

The architecture of the RNN was devised to improve the information storage capacity of the input data to overcome the shortcomings of the existing DNN, which cannot remember the previous data after converting them to predicted values [19]. To achieve this, the RNN adds a cyclic connection to each neuron based on the structure of the DNN; therefore, this architecture allows processing of time-series data in which dependencies exist between the data in the order of the input. In

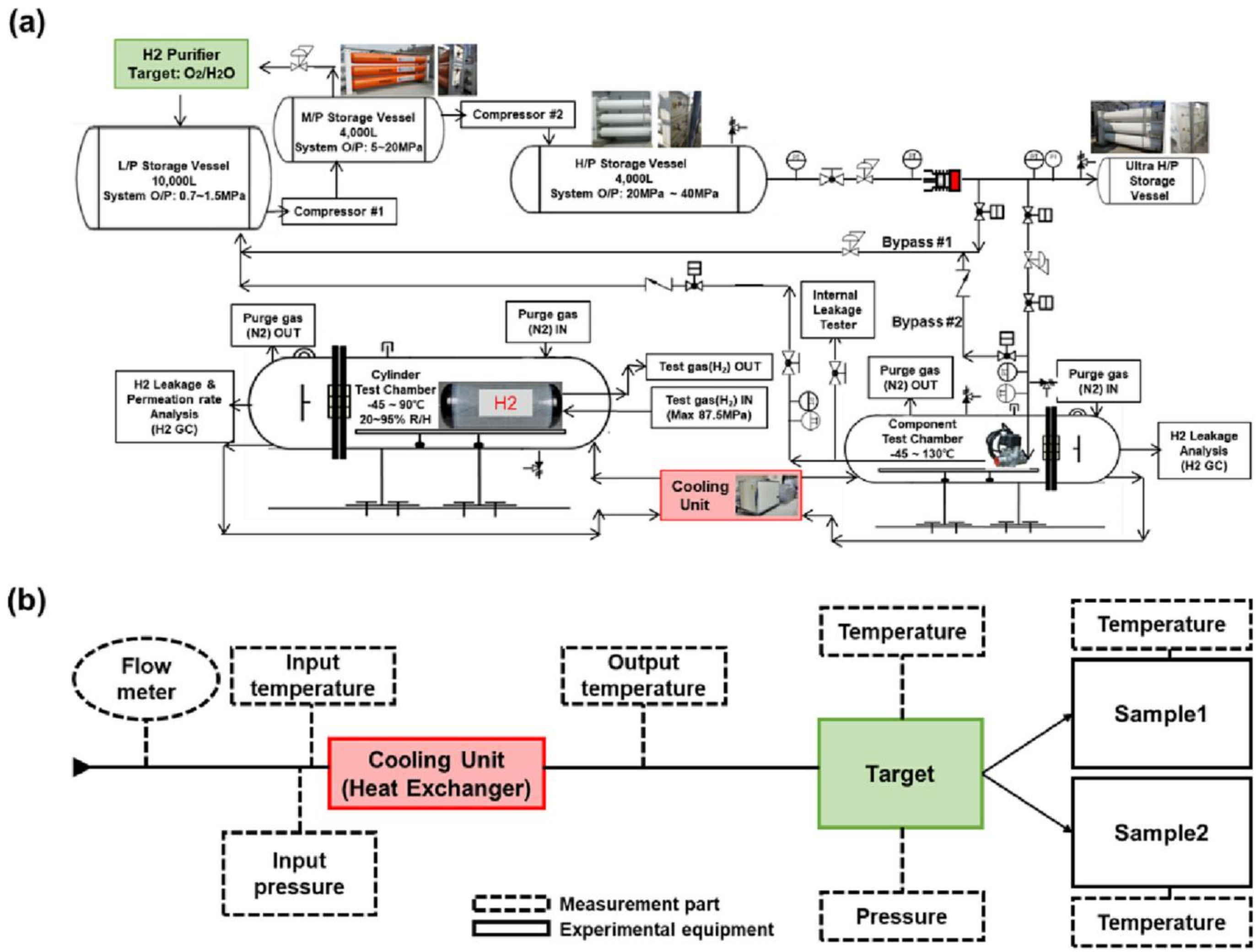


**Fig. 1.** Systematic diagram of compressed hydrogen system | (a) Total facility for hydrogen stability assessment, (b) Simplified schematic of the test facility for the experimental data used for modeling.

**Table 1**
Details of the measuring device.

| Part | Model | Range |
|---|---|---|
| Coriolis flow meter | CA004G13 - 22AA1134 | 0–5 kg/min, ±0.05 % of full scale |
| Input pressure | P2584A | 0–160 MPa, ±1 % |
| Input temperature | R221(S) | −100 to 450 °C |
| Output temperature | NTC thermistor | −55 to 200 °C |
| Target temperature | R221(S) + A6061 | −100 to 450 °C |
| Target pressure | SIG H A - M D G | 0–150 MPa, ±1 % |
| Sample (1) temperature | NTC thermistor | −55 to 200 °C |
| Sample (2) temperature | NTC thermistor | −55 to 200 °C |

**Table 2**
Test case for each experimental environment condition.

| Case | Ambient temperature [°C] | Time [s] | Flow conditions [g/s] |
|---|---|---|---|
| 1 | 19 | 0–264 | 5–6 |
| 2 | 19 | 0–216 | 5–6 |
| 3 | 19 | 0–230 | 5–6 |
| 4 | 12 | 0–482 | 3–5 |
| 5 | 12 | 0–177 | 0–80 |
| 6 | 5 | 0–178 | 0–80 |

the RNN architecture, the cycles are repeated in temporal order. The outputs of the hidden layer at the previous time instant ($h_{t-1}$) are back-fed to the neurons for calculating the outputs of the hidden layer at the current time instant ($h_t$) combined with the current data ($x_t$) and weights (W). Thus, the RNN stores prior information in the hidden layer. The structure of the RNN is briefly expressed as an equation as follows.

$$h_t = tanh\left[W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right] \tag{1}$$

However, conventional RNNs, such as that in Fig. 2(b), have vanishing/exploding gradient that abnormally decrease or increase the gradients when learning data over long periods of time based on the backpropagation algorithm [20]. To resolve this issue, the long short-term memory (LSTM) [21] network was proposed as a type of RNN, as shown in Fig. 2(c). LSTMs are explicitly designed to avoid the problem of long-term dependencies. In the LSTM, a forget gate, an input gate, and an output gate are added so that the memory cell is implemented in the neurons of the hidden layer. The structure of the LSTM can be briefly expressed as Eqs. (2) and (3). An input gate ($i_t$), a forget gate ($f_t$), and an output gate ($o_t$) are calculated using the sigmoid function (σ) with the previous outputs of the hidden layer ($h_{t-1}$) and current data ($x_t$). The input gate ($i_t$) decides the current data to be stored and combined with the output of the current gate ($g_t$), while the forget gate ($f_t$) decides which of the previous cell states ($c_{t-1}$) are to be discarded. Thus, the current cell state update ($c_t$) is performed, and the updated cell states are

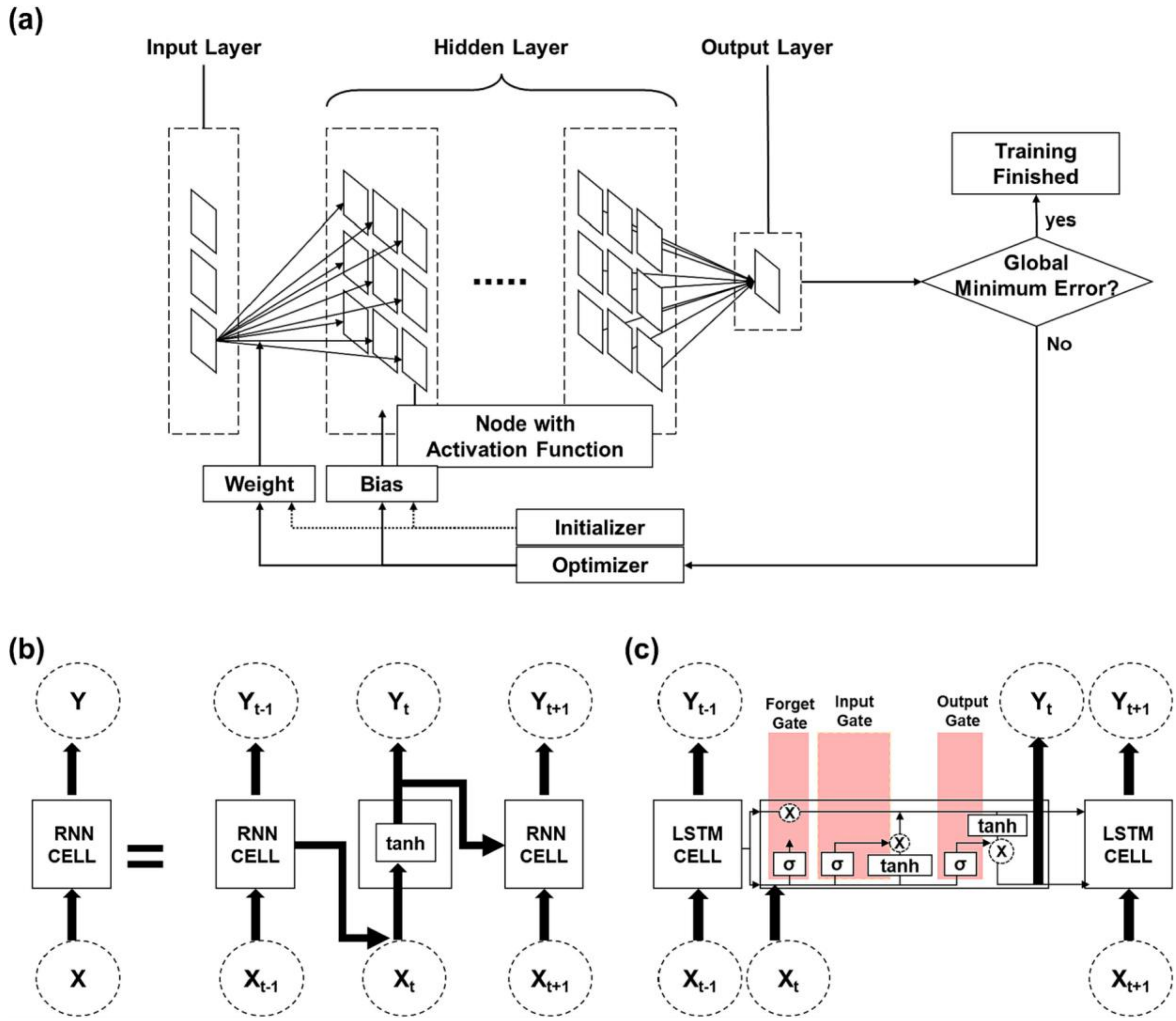


**Fig. 2.** Architectures of the neural network models used in this study | (a) DNN, (b) conventional RNN, and (c) long short-term memory (LSTM)-based RNN.

used in the hyperbolic tangent function (tanh) to calculate the output values ($h_t$) at the output gate ($o_t$). Hence, LSTM has been adopted for RNN modeling.

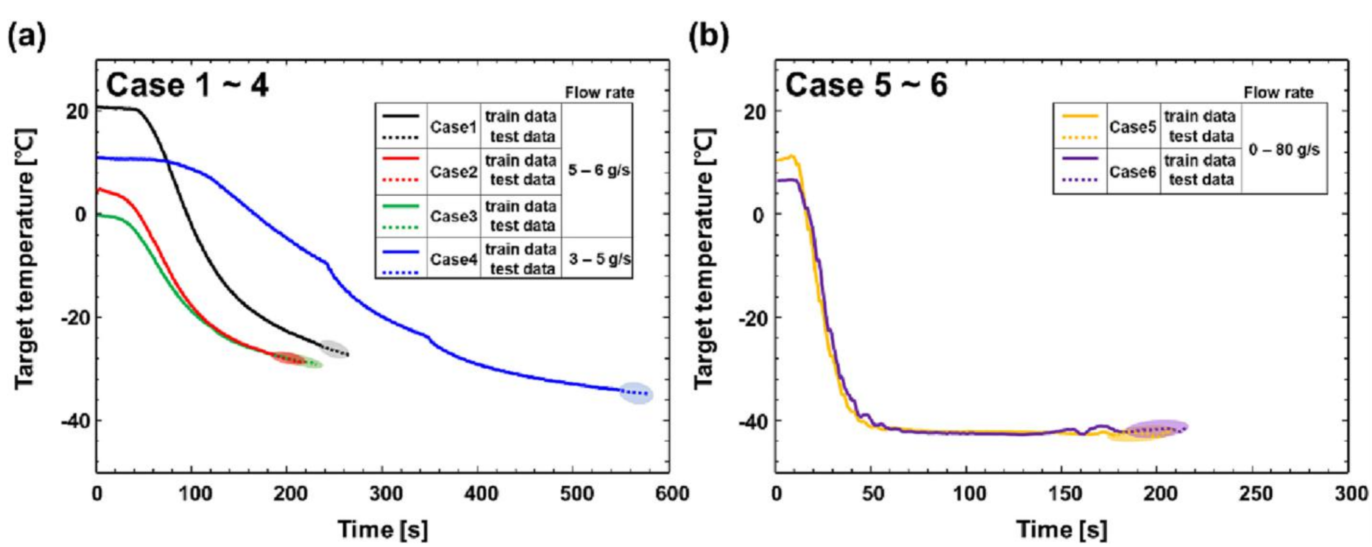


**Fig. 3.** The change of target temperature over time for each case | (a) Cases 1–4 with low and steady state flow rates, (b) Cases 5 & 6 with high and variable flow rates.

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} W \tag{2}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{3-a}$$

$$h_t = o_t \cdot tanh(c_t) \tag{3-b}$$

### 2.2.2. System modeling process

As shown in Table 2, experimental data were obtained for each case. The test data are separated at the final 30 s of operation time for each case, and 70 % of the remaining data are randomly used for the training, while 30 % of data are used for validation. As shown in Fig. 3, Cases 1–4 are tested at low and steady state flow rates, while Cases 5 and 6 are tested at high and variable flow rates, showing rapid temperature decreases. Also, the test data area used in each case is highlighted in Fig. 4. The modeling performances using DNN and RNN were compared by predicting these target temperatures at the final 30 s of operation time.

For DNN modeling, to develop a general useful model, as shown in Fig. 4, the entire data of all cases were used with ambient temperature labeling for each case. The RNN was constructed individually for each case since the RNN can only be developed from continuous time-series data.

All ANN models were trained using TensorFlow (version 2.4.3) and Numpy (version 1.19.2). These two tools enabled us to handle and train the multidimensional matrix arrays based on Python (version 3.9). For both models, min-max normalization was performed during data preprocessing [22]. For the DNN modeling, the leaky rectified linear unit (LReLU) [23], which is a modified ReLU [24] function to prevent convergence to zero, was used as the activation function, which is expressed as

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0.2x & \phantom{for } x < 0 \end{cases} \tag{4}$$

Initialization was used with the He method for the weights, which has been known to have good fit for the ReLU series function [25], and the bias is initialized to zero. The LReLU was used as the activation function in the hidden layer with the dropout method [26], and the identity function was used in the output layer. For LSTM modeling, the existing structure in TensorFlow was used. Both models were updated with the widely used Adam optimizer [27] during training, which is a combination of AdaGrad [28] and RMSprop [29]. The early stopping method, which is also a widely used convention in neural networks, was used to minimize the number of iterations while avoiding overfitting of the training data. This method assesses whether the learning process has converged and is complete.

### 2.3. Hyperparameter optimization

Hyperparameter optimization of a model plays an important role in determining the modeling performance. For both models, the batch size, learning rate, number of nodes, and number of layers were selected as the hyperparameters to be optimized, and the value of the dropout was additionally selected only for the DNN model. To reduce the computational cost, the optimization and activation functions were fixed as described above. Hyperopt based on the Bayesian optimization method [30] was used for optimization; since this method reflects the pre-values in the optimal value search, it can reduce costs compared to the grid search approach, which checks all parameters, and increase accuracy compared to random search. To select a model with excellent performance, the K-fold cross validation was performed to compensate for the limitations with the amount of data [31,32]. After separating the test data, K-fold cross-validation with K = 10 was used to train and validate the data.

For DNN modeling, the batch size, learning rate, number of nodes, number of layers, and dropout were selected as the hyperparameter targets to be optimized; however, as the number of hyperparameters increase, it is known to be inefficient to find the optimal values owing to the variety in the number of cases [33]. Therefore, the optimal range for the number of layers was first selected to limit the range of depths of the deep network that affects modeling performance, and good results were obtained when the number of layers was between 10 and 14. When the numbers of layers were 10, 12, and 14, about 10 optimal hyperparameter sets were derived using Bayesian optimization, and a total of six sets were selected as the candidates by selecting two sets in each layer that had the best performances. Finally, to exclude the bias for the data set, K-fold validation was conducted for the six hyperparameter set. The performance of predictive accuracy of each model was determined using the average value of the MAE, which can be obtained as follows:

$$MAE(\%) = \frac{1}{n} \sum_{n} \frac{|f_{exp} - f_{pred}|}{f_{exp}} \times 100 \tag{5}$$

The first set was selected with the lowest MAE of the K-fold validation data (MAE = 3.9 × $10^{-4}$ %). Information on the six optimized hyperparameter sets is shown in Table 3.

For RNN modeling, the batch size, learning rate, number of nodes, and number of layers were selected as the targets for the hyperparameters to be optimized, and when the optimal range for the number

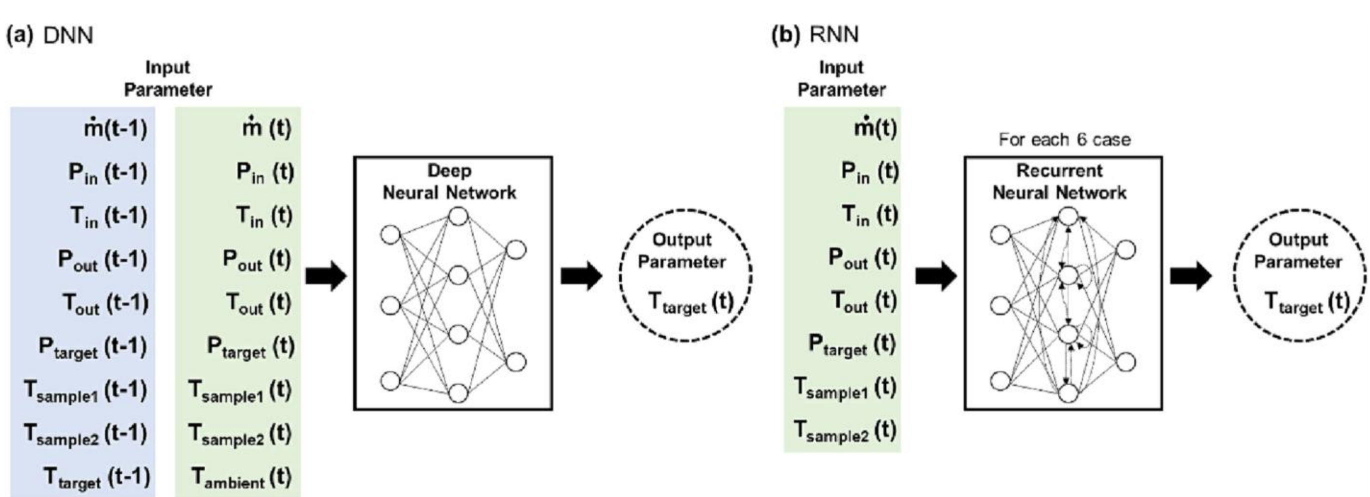


**Fig. 4.** Input and output parameters for each model | (a) Universal DNN modeling with 18 input parameters containing information of previous time (t-1), (b) 6 RNN modeling with 8 input parameters without previous information.

**Table 3**
Best hyperparameter sets for DNN modeling.

| Set | | 1(best) | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| MAE of validation data [%] | | 3.91E-04 | 5.07E-04 | 4.01E-04 | 1.41E-02 | 1.37E-03 | 6.69E-04 |
| Layer | | 10 | 10 | 12 | 12 | 14 | 14 |
| Batch size | | 2 | 4 | 8 | 16 | 8 | 8 |
| Dropout | | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.3 |
| Learning rate | | 0.001 | 0.005 | 0.0005 | 0.0005 | 0.001 | 0.001 |
| Nodes | 1 | 32 | 128 | 128 | 128 | 64 | 128 |
| | 2 | 32 | 128 | 64 | 32 | 4 | 4 |
| | 3 | 4 | 128 | 64 | 8 | 8 | 32 |
| | 4 | 16 | 32 | 32 | 64 | 4 | 8 |
| | 5 | 32 | 4 | 128 | 32 | 4 | 64 |
| | 6 | 16 | 128 | 8 | 32 | 16 | 32 |
| | 7 | 32 | 128 | 32 | 4 | 8 | 8 |
| | 8 | 16 | 128 | 32 | 4 | 8 | 64 |
| | 9 | 8 | 32 | 64 | 4 | 8 | 64 |
| | 10 | 64 | 64 | 16 | 32 | 16 | 8 |
| | 11 | | | 64 | 32 | 32 | 16 |
| | 12 | | | 8 | 64 | 16 | 16 |
| | 13 | | | | | 16 | 4 |
| | 14 | | | | | 32 | 8 |

of layers was first limited in the same way as that of the DNN model, the results were between 2 and 4. When the numbers of layers were 2, 3, and 4, the best performing hyperparameter set was used for the RNN model in each case using Bayesian optimization. For each case, the finally selected hyperparameter set is shown in Table 4.

## 3. Results and discussion

### 3.1. Summary of prediction results with the DNN and RNN models

Fig. 5 shows the summarized MAEs obtained using the DNN and RNN for each test case in Table 2. Overall, the DNN (0.41 %–1.71 %) exhibited better predictive accuracies in terms of MAE compared to the RNN (1.05 %–3.11 %). The DNN predicted better than the RNN for all cases except Case 1. In particular, for Cases 5 and 6, the MAEs of the DNN (0.57 %–0.71 %) were substantially lower (by approximately three times) than those of the RNN (3.10 %–3.11 %). Our DNN model outperformed one that was utilized in a similar dynamic system in a previous study [34]. In particular, the previous study reported an MAE of 1.1 %–5 % for the outlet temperature in the absorption chiller, whereas our DNN model achieved higher predictive accuracy. Moreover, our RNN model also exhibited better accuracy compared to previous research that used a many-to-one structure like ours. A study that predicted temperature and humidity in a solar greenhouse showed MAE values of 3.9 % and 4.3 %, respectively [35]. The differences between Cases 1 to 4 and Cases 5 and 6 have effects on the models' prediction performances. Cases 1 to 4 had relatively small and constant flow rates, while Cases 5 and 6 had fluctuating flow rates. In the following Sections 3.2 and 3.3, more detailed prediction performances are presented by comparing the temporal temperature predictions for all test cases.

### 3.2. Temperature predictions by the DNN and RNN

Fig. 6 shows the comparisons of the temporal temperature predictions at flow rates of 5–6 g/s and 3–5 g/s. In Fig. 6, Cases 2, 3, and 4 are seen to have higher MAEs with the RNN than the DNN, while a lower MAE is seen for the RNN than DNN for Case 1. In Fig. 6(a), for Case 1, the predicted temperature of the RNN is higher than the experimental value, and it is the only case where RNN has better performance than DNN. On the other hand, RNN underpredicted the temperatures for Cases 2 and 3 than the experimental values, while DNN overpredicted the temperatures. Case 4 shows that both RNN and DNN predicted lower temperatures. The reason for the worse performance of DNN than RNN in Case 1 is related to the method of modeling the DNN. Owing to the characteristics of the DNN modeling, which includes all cases in one model, the DNN model has no choice but to predict similar trends with matching parameters. Cases 1, 2, and 3 were tested at the same ambient temperature of 19 °C, but the initial temperature of the target tank varies from 19 °C for Case 1 to 0 °C for Case 3. Therefore, even if the DNN considers the effects of the initial temperature of the target, the training data with the same ambient temperatures (Cases 2 and 3) result in a lower predicted temperature for Case 1.

Unlike the RNN that is built with successive time series, the DNN has the advantage that the total data can be learned at once, regardless of continuity, and used as a universal model. However, this could result in larger errors if discontinuous time series are included under the same environmental conditions with different initial temperatures and pressures of the target tank.

Fig. 7 shows the comparisons of temperature predictions for flow rates of 0–80 g/s. In Fig. 7(a) and (b), the performance of the DNN is much better than that of the RNN. Because the DNN predicts the target temperature of the current time (t) based on the input parameter that has information about the target temperature at the previous time (t-1), it exhibits excellent temperature prediction even in the presence of flow-rate fluctuations. On the other hand, the RNN shows good agreement with the temporal temperature changes when the flow rate changes. However, the temperature change in the RNN appears to be different because the flow rate change is not reflected immediately owing to the relatively large heat capacity of the system. This trend can be observed during 18–22 s in Fig. 7(a) when the experienced temperature decreases from −42.5 °C to −43.5 °C and the predicted temperature of the RNN increases from −41.5 °C to −40 °C.

### 3.3. Heat transfer rate predictions by the DNN and RNN

Fig. 8(a–d) shows the comparisons of the experimentally determined heat transfer rates based on predictions by the DNN and RNN models for four constant flow-rate conditions. The calculated heat transfer rate is determined using the inlet temperature and mass flow rate based on the predicted target temperature. The heat transfer rate is calculated by multiplying the difference between the inlet and target temperature with the mass flow rate. Since the total heat transfer rate of a system is a function of its mass flow rate, most changes in the heat transfer rate follow a trend with respect to the mass flow rate. As with the temperature predictions, the DNN showed good performances with excellent MAEs for all cases except Case 1. Contrary to the concerns that the

**Table 4**
Best hyperparameter sets of the RNN model for each of six cases.

| Case | | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Layer | | 3 | 4 | 4 | 4 | 3 | 3 |
| Batch size | | 2 | 14 | 18 | 6 | 6 | 6 |
| Learning rate | | 0.00085 | 0.001 | 0.0009 | 0.002 | 0.0009 | 0.001 |
| Nodes | 1 | 28 | 256 | 256 | 128 | 36 | 256 |
| | 2 | 28 | 256 | 256 | 128 | 36 | 256 |
| | 3 | 22 | 256 | 256 | 128 | 46 | 22 |
| | 4 | | 22 | 18 | 8 | | |

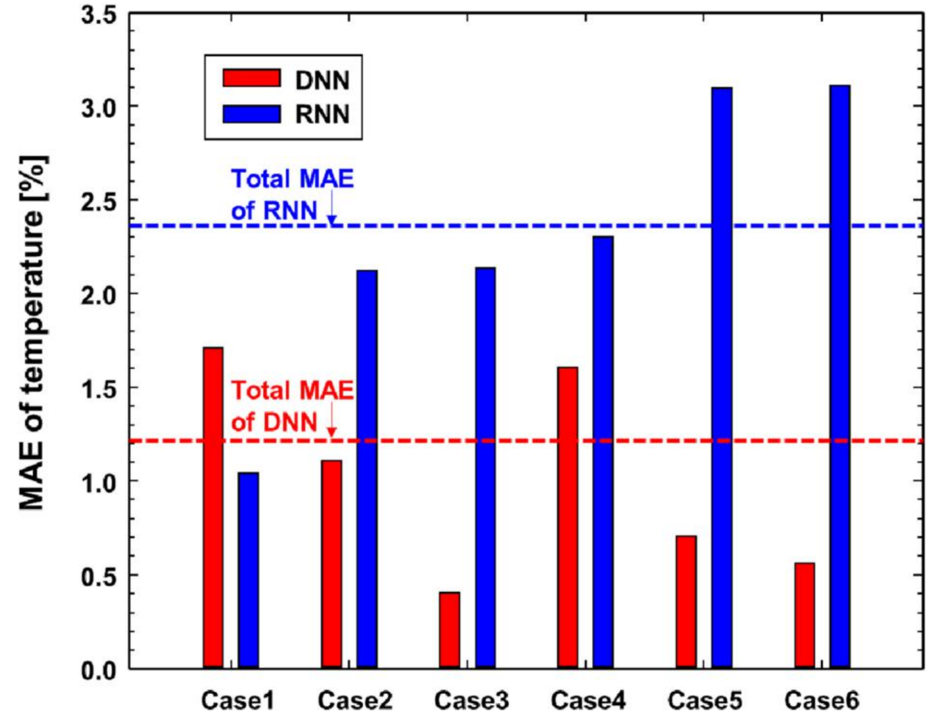


**Fig. 5.** Comparison of MAEs of DNN and RNN models for each case.

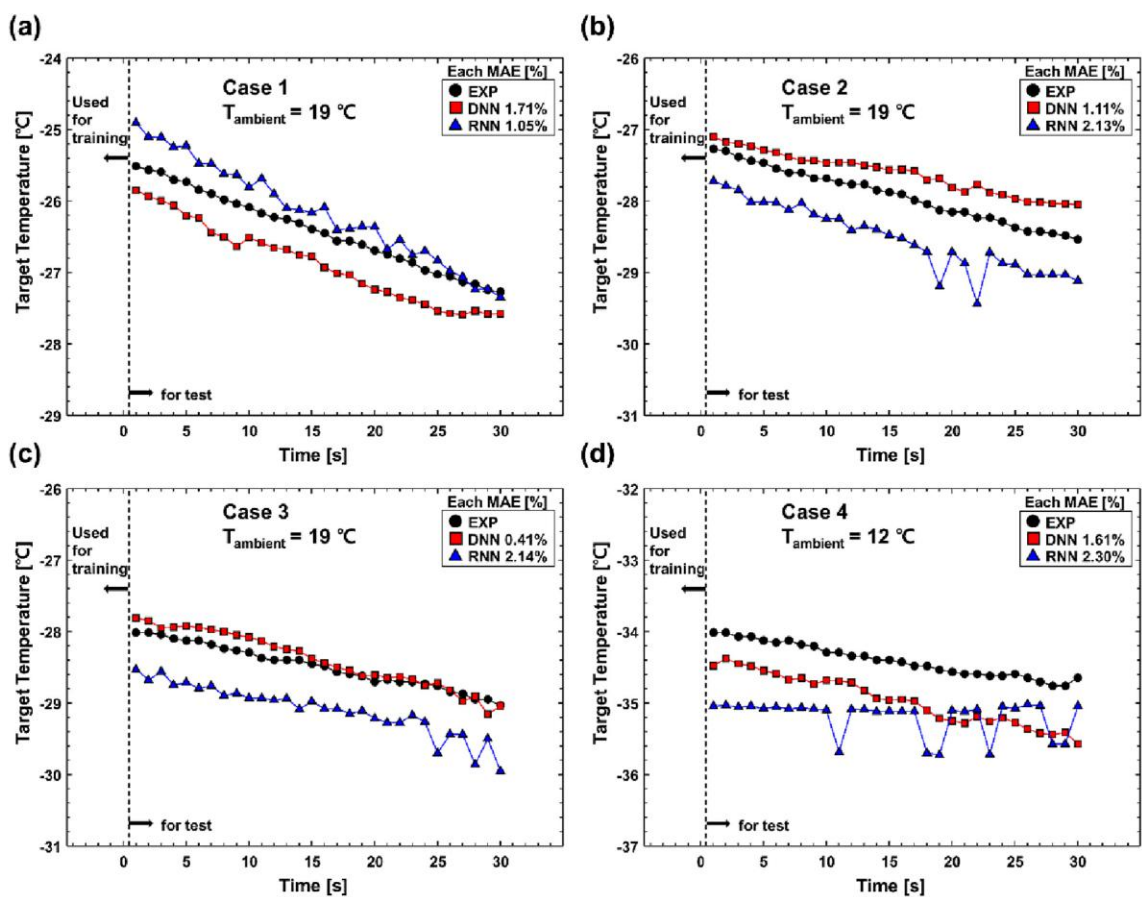


**Fig. 6.** Comparisons of target temperature predictions with experimental test data at low flow rates | (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4.

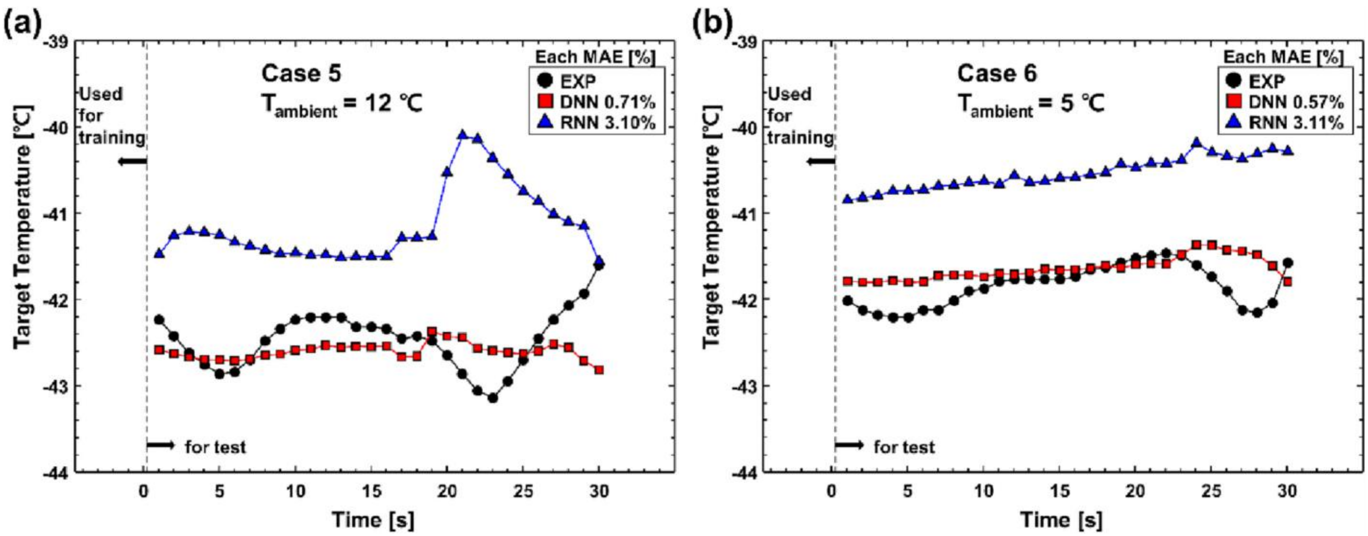


**Fig. 7.** Comparisons of target temperature predictions with experimental test data at high flow rates | (a) Case 5, (b) Case 6.

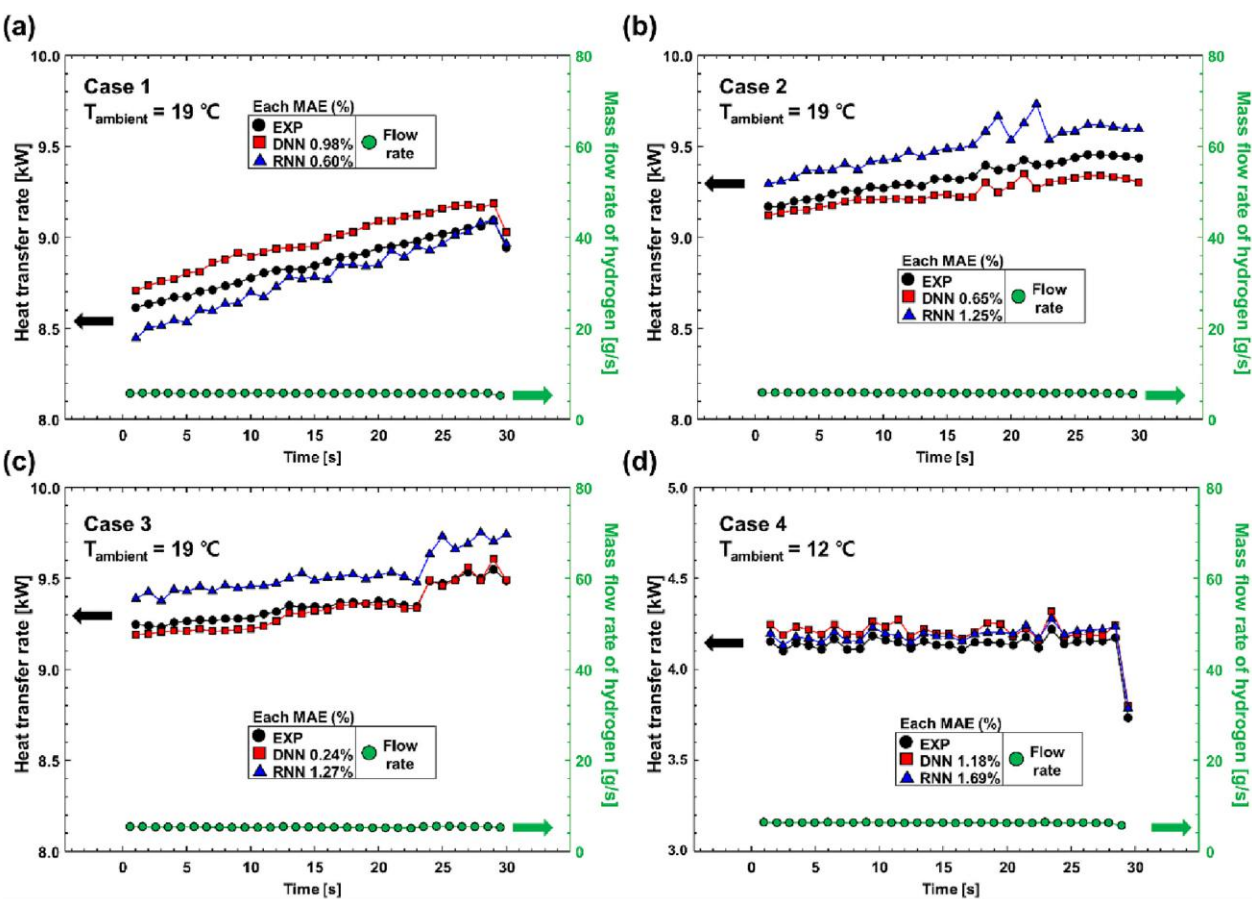


**Fig. 8.** Comparisons of temporal variations of experimentally determined heat transfer rate with predictions by the DNN and RNN and mass flow rate | (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4.

performance of the RNN would deteriorate as the time point of the test data is farther away than the time point of the training data, the MAE in Case 1 decreased over time and the MAEs for Cases 2, 3, and 4 were constant over time.

Fig. 9 shows the comparisons of the temporal variations of the experimentally determined heat transfer coefficients based on predictions by the DNN and RNN models under variable flow rate conditions. Owing to the flow rate change, the total heat transfer rates abruptly changed from 0 to 30 kW with time for Cases 5 and 6. As shown in Fig. 9, the DNN shows good agreement with the experimentally measured target temperatures for Cases 5 and 6, and the heat transfer rates are also predicted well by the DNN for these cases. The RNN exhibits a relatively high MAE at the start of the test but begins to decrease after 20 s as the flow rate decreases. Both the DNN and RNN are thus

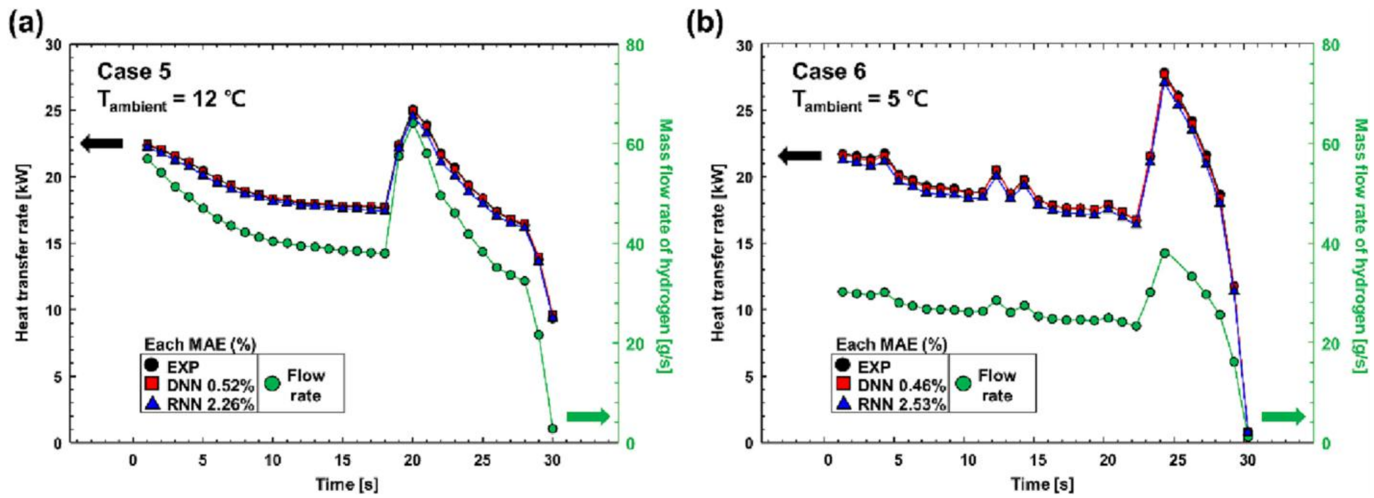


**Fig. 9.** Comparisons of temporal variations of experimentally determined heat transfer rate with predictions by the DNN and RNN and mass flow rate | (a) Case 5, (b) Case 6.

able to predict the total heat transfer rates based on the target prediction temperatures within 3 % of MAEs for all cases.

## 4. Conclusions

This study developed DNN and RNN models to predict the ultra-low temperature of hydrogen passing through a heat exchanger under 6 different operating environments. The performance of DNN prediction of the target temperature is better than that of the RNN for all cases except Case 1. The performance of RNN predictions of the target temperatures in Cases 5 and 6, which operated at variable flow rates, was worse than those in Cases 1 to 4, which operated at constant flow rates.

DNN modeling shows more advantageous performance with a variable flow rate in that it predicts the target temperature at the current time (t) by considering the target temperature at the previous time (t-1) as an input parameter. In other words, DNN modeling has the disadvantage that it cannot predict the target temperature at the current time (t) without that at the previous time (t-1), and RNN modeling can be used to overcome this limitation. The RNN can predict the target temperature at the current time (t) under different operating conditions without the influence of the value at the previous time (t-1).

In summary, both DNN and RNN models can predict the target temperatures of hydrogen within 3.11 % MAE and heat transfer rates of the heat exchanger within 2.53 % MAEs. DNN modeling is recommended as a universal method for predicting the temperature when previous data are known, and RNN modeling is recommended for predicting the temperature for successive actions in a compressed hydrogen system using a cooling heat exchanger.

The proposed approach could be highly beneficial for compressed-hydrogen systems by providing a reliable means to estimate hydrogen temperature before designing the entire system. This allows users to have accurate information about their setup. Additionally, the quick prediction time of the ANN model is advantageous for engineers who need to compare different solutions in various environments, eliminating the need for detailed physical dynamic models that are time-consuming to use for simulations. However, to use this model universally, adding experiments for ambient temperatures ranging from −30 °C to 40 °C is necessary to ensure the usefulness of the modeling. In addition, to further improve the proposed methodology, future works will focus on performing comparative analyses across different types of equipment and incorporating more advanced deep-learning techniques for modeling updates.

### Declaration of competing interest

None.

### Data availability

Data will be made available on request.

### Acknowledgments

This work was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant no. 22HBST-C158067-03), the Industrial Technology Innovation Program (No. 20007058, Development of safe and comfortable human augmentation hybrid robot suit) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), and the Chung-Ang University Graduate Research Scholarship in 2022.

Haeeun Lee would like to express her gratitude to the Hyundai Motor Chung Mong-Koo Foundation.

### References

[1] E. Dabirian, A. Hajipour, A.A. Mehrizi, C. Karaman, F. Karimi, P. Loke-Show, O. Karaman, Nanoparticles application on fuel production from biological resources: a review, Fuel. 331 (2023), 125682, https://doi.org/10.1016/j.fuel.2022.125682.

[2] H. Jouhara, N. Khordehgah, S. Almahmoud, B. Delpech, A. Chauhan, S.A. Tassou, Waste heat recovery technologies and applications, Therm. Sci. Eng. Prog. 6 (2018) 268–289, https://doi.org/10.1016/j.tsep.2018.04.017.

[3] J. Tollefson, Hydrogen vehicles: fuel of the future? Nature. 464 (2010) 1262–1264, https://doi.org/10.1038/4641262a.

[4] A. Züttel, Hydrogen storage methods, Naturwissenschaften. 91 (2004) 157–172, https://doi.org/10.1007/s00114-004-0516-x.

[5] A. Chibani, G. Mecheri, A. Dehane, S. Merouani, I. Ferhoune, Performance improvement of adsorptive hydrogen storage on activated carbon: effects of phase change material and inconstant mass flow rate, J. Energy Storage. 56 (2022), 105930, https://doi.org/10.1016/j.est.2022.105930.

[6] J. Ye, Z. Zhao, J. Cui, Z. Hua, W. Peng, P. Jiang, Transient flow behaviors of the check valve with different spool-head angle in high-pressure hydrogen storage systems, J. Energy Storage. 46 (2022), 103761, https://doi.org/10.1016/j.est.2021.103761.

[7] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (1989) 303–314, https://doi.org/10.1007/BF02551274.S.A.

[8] Kalogirou, Artificial neural networks in renewable energy systems applications: a review, Renew. Sust. Energ. Rev. 5 (2001) 373–401, https://doi.org/10.1016/S1364-0321(01)00006-5.

[9] Y. Islamoglu, A new approach for the prediction of the heat transfer rate of the wire-on-tube type heat exchanger-use of an artificial neural network model, Appl. Therm. Eng. 23 (2003) 243–249, https://doi.org/10.1016/S1359-4311(02)00155-2.

[10] H. Lee, M. Kang, K. Wook, C.R. Kharangate, S. Lee, M. Iyengar, C. Malone, M. Asheghi, K.E. Goodson, H. Lee, An artificial neural network model for predicting frictional pressure drop in micro-pin fin heat sink, Appl. Therm. Eng. 194 (2021), 117012, https://doi.org/10.1016/j.applthermaleng.2021.117012.
[11] K. Kim, H. Lee, M. Kang, G. Lee, K. Jung, C.R. Kharangate, M. Asheghi, K. E. Goodson, H. Lee, A machine learning approach for predicting heat transfer characteristics in micro-pin fin heat sinks, Int. J. Heat Mass Transf. 194 (2022), https://doi.org/10.1016/j.ijheatmasstransfer.2022.123087.
[12] S. Motahar, Experimental study and ANN-based prediction of melting heat transfer in a uniform heat flux PCM enclosure, J. Energy Storage. 30 (2020), 101535, https://doi.org/10.1016/j.est.2020.101535.
[13] A. Behzadi, S.M. Alirahmi, H. Yu, S. Sadrizadeh, An efficient renewable hybridization based on hydrogen storage for peak demand reduction: a rule-based energy control and optimization using machine learning techniques, J. Energy Storage. 57 (2023), 106168, https://doi.org/10.1016/j.est.2022.106168.
[14] E. Cho, H. Lee, M. Kang, D. Jung, G. Lee, S. Lee, C.R. Kharangate, H. Ha, S. Huh, H. Lee, A neural network model for free-falling condensation heat transfer in the presence of non-condensable gases, Int. J. Therm. Sci. 171 (2022), 107202, https://doi.org/10.1016/j.ijthermalsci.2021.107202.
[15] C. Karaman, O. Karaman, P.L. Show, H. Karimi-Maleh, N. Zare, Congo red dye removal from aqueous environment by cationic surfactant modified-biomass derived carbon: equilibrium, kinetic, and thermodynamic modeling, and forecasting via artificial neural network approach, Chemosphere. 290 (2022), 133346, https://doi.org/10.1016/j.chemosphere.2021.133346.
[16] O. Karaman, Three-dimensional graphene network supported nickel-cobalt bimetallic alloy nanocatalyst for hydrogen production by hydrolysis of sodium borohydride and developing of an artificial neural network modeling to forecast hydrogen production rate, Chem. Eng. Res. Des. 181 (2022) 321–330, https://doi.org/10.1016/j.cherd.2022.03.028.
[17] Z. Xi, R. Wang, Y. Fu, C. Mi, Accurate and reliable state of charge estimation of lithium ion batteries using time-delayed recurrent neural networks through the identification of overexcited neurons, Appl. Energy 305 (2022), 117962, https://doi.org/10.1016/j.apenergy.2021.117962.
[18] G. Liu, L. Guo, C. Liu, Q. Wu, Evaluation of different calibration equations for NTC thermistor applied to high-precision temperature measurement, Meas. J. Int. Meas. Confed. 120 (2018) 21–27, https://doi.org/10.1016/j.measurement.2018.02.007.
[19] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent Neural Network Regularization, (2014). http://arxiv.org/abs/1409.2329.
[20] L.R.Medsker, and L. C. Jain., Recurrent neural networks, Design and Appl. 5 (2001) 64–67. https://dl.acm.org/doi/abs/10.5555/553011.
[21] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Phys. D Nonlinear Phenom. 404 (2020), https://doi.org/10.1016/j.physd.2019.132306.
[22] S.G.K. Patro, K.K. Sahu, Normalization: {A} Preprocessing Stage, CoRR. abs/1503.0 (2015). http://arxiv.org/abs/1503.06462.
[23] M.J. Brown, L.A. Hutchinson, M.J. Rainbow, K.J. Deluzio, A.R. De Asha, A comparison of self-selected walking speeds and walking speed variability when data are collected during repeated discrete trials and during continuous walking, J. Appl. Biomech. 33 (2017) 384–387. doi:10.1123/jab.2016-0355.
[24] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, ICML Work, Deep Learn. Audio, Speech Lang. Process. 28 (2013).
[25] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, Proc. IEEE Int. Conf. Comput. Vis. 2015 Inter (2015) 1026–1034. doi:https://doi.org/10.1109/ICCV.2015.123.
[26] P. Baldi, P. Sadowski, The dropout learning algorithm, Artif. Intell. 210 (2014) 78–122, https://doi.org/10.1016/j.artint.2014.02.004.
[27] Kingma, J.L. Ba, Adam: A method for stochastic optimization, 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. (2015) 1–15.
[28] J.C. Duchi, P.L. Bartlett, M.J. Wainwright, Randomized smoothing for (parallel) stochastic optimization, Proc. IEEE Conf. Decis. Control. 12 (2012) 5442–5444, https://doi.org/10.1109/CDC.2012.6426698.
[29] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks for Machine Learning 4 (2) (2012) 26–31.
[30] H. Snoek, R. Larochelle, Adams, practical Bayesian optimization of machine learning algorithms, Adv. Neural Inf. Process. Syst. 4 (2012).
[31] U.M. Braga-Neto, E.R. Dougherty, Is cross-validation valid for small-sample microarray classification? Bioinformatics. 20 (2004) 374–380, https://doi.org/10.1093/bioinformatics/btg419.
[32] T. Fushiki, Estimation of prediction error by using K-fold cross-validation, Stat. Comput. 21 (2011) 137–146, https://doi.org/10.1007/s11222-009-9153-8.
[33] R. Bardenet, M. Brendel, B. Kégl, M. Sebag, Collaborative hyperparameter tuning, in: S. Dasgupta, D. McAllester (Eds.), Proc. 30th Int. Conf. Mach. Learn., PMLR, Atlanta, Georgia, USA, 2013: pp. 199–207. https://proceedings.mlr.press/v28/bardenet13.html.
[34] A. Lazrak, F. Boudehenn, S. Bonnot, G. Fraisse, A. Leconte, P. Papillon, B. Souyri, Development of a dynamic artificial neural network model of an absorption chiller and its experimental validation, Renew. Energy 86 (2016) 1009–1022, https://doi.org/10.1016/j.renene.2015.09.023.
[35] W. Hongkang, L. Li, W. Yong, M. Fanjia, W. Haihua, N.A. Sigrimis, Recurrent neural network model for prediction of microclimate in solar greenhouse, IFAC-PapersOnLine. 51 (2018) 790–795, https://doi.org/10.1016/j.ifacol.2018.08.099.