Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2022, 2022:98
https://doi.org/10.28919/cmbn/7655
ISSN: 2052-2541

COMPARISON OF DEEP LEARNING SEQUENCE-TO-SEQUENCE MODELS IN PREDICTING INDOOR TEMPERATURE AND HUMIDITY IN SOLAR DRYER DOME

KARLI EKA SETIAWAN[1,*], GREGORIUS NATANAEL ELWIREHARDJA[2], BENS PARDAMEAN[1,2]

[1]Computer Science Department, BINUS Graduate Program, Master of Computer Science Program, Bina Nusantara University Jakarta, Indonesia 11480

[2]Bioinformatics and Data Science Research Center, Bina Nusantara University, Jakarta, Indonesia 11480

Abstract: Solar Dryer Dome (SDD), which is an agriculture facility for preserving and drying agriculture products, needs an intelligent system for predicting future indoor climate conditions, including temperature and humidity. An accurate indoor climate prediction can help to control its indoor climate conditions by efficiently scheduling its actuators, which include fans, heaters, and dehumidifiers that consume a lot of electricity. This research implemented deep learning architectures to predict future indoor climate conditions such as indoor temperature and indoor humidity using a dataset generated from the SDD facility in Sumedang, Indonesia. This research compared adapted sequenced baseline architectures with sequence-to-sequence (seq2seq) or encoder-decoder architectures in predicting sequence time series data as the input and output of both architecture models which are built based on Recurrent Neural Network (RNN) layers such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). The result shows that the adapted sequence baseline model using GRU is the best model, whereas seq2seq models yield bigger Mean Absolute

*Corresponding author
E-mail address: karli.setiawan@binus.ac.id
Received August 06, 2022

Error (MAE) values by almost ten times. Overall, all the proposed deep learning models are categorized as extremely strong with $R^2 \geq 0.99$.

Keywords: deep learning; solar dryer dome; sequence-to-sequence prediction; indoor climate prediction.

2010 AMS Subject Classification: 68T05, 62H20, 97R40, 97R30.

1. INTRODUCTION

Indonesia has implemented the Indonesia Agriculture 4.0 programs, which means that the agriculture system should consist of Artificial Intelligence (AI) or Machine Learning (ML), the Internet of Things (IoT), and cyber-physical systems. One of those programs is Smart Dome 4.0, a low-cost, eco-friendly, and sophisticated program to support Indonesian farmers in saving their agricultural products [1]. The purpose of building a Solar Dryer Dome (SDD) is for food preservation and maintaining the product's nutritional content because agricultural products require a long time to process before they are delivered to consumers [2]. SDD overcomes the many shortcomings of traditional drying methods under the sun in the outdoors, such as longer drying processes, potential rain, dust impact, bird and other flying animal droppings, the growth of fungi, inappropriate humidity, and color change.

One weakness of SDD is the need for a power source for running the system continuously to provide suitable indoor environmental conditions with a constant supply of electricity for operating the actuators such as fans, heating systems, and dehumidifiers [1] [3] [4]. SDD uses green energy by collecting solar energy using a solar panel during the day and storing it in a battery for use at night. Indonesia, a country with two seasons, has various solar radiation distributions, so it can become a problem for SDD for solar energy absorption [5]. When the weather is dark and rainy throughout the day, it also becomes a problem. In a mountain area, dramatic weather changes also affect indoor SDD significantly. Another study on SDD concludes that controlling indoor climate by increasing indoor temperature and decreasing indoor humidity consumes the most power [6]. It makes predicting environmental parameters for scheduling the actuators an important thing for SDD. The application of the actuator scheduling can reduce SSD power consumption by using energy sparingly.

Predicting indoor climate for controlling SDD environmental conditions in order to achieve power consumption efficiency and the best quality of dried agricultural products is one of the most important and difficult tasks to perform for SDD [7]. Deep learning (DL), a method that learns the distribution of the data to be modeled automatically, can be applied to address these challenges in the agriculture sector, especially DL with RNN [8] [9]. Many reports show how amazing RNN can solve various challenges where the data is sequential [10] [11]. RNN can learn historical information in time series data with the aim of predicting future results [12]. Later, special improved RNNs such as LSTM and GRU appear, which are intended for long-term learning. [13]. Sequence-to-sequence (seq2seq) or encoder-decoder is one of the deep learning architectures which is popularly implemented in Natural Language Processing (NLP), which can output sequence data with sequence data input [14]. Since the input and output data are sequence, this research compared both the adapted sequence baseline architecture and the seq2seq architecture, which applied RNN layers on it, so the proposed 4 models are adapted sequence baseline with stacked GRU, adapted sequence baseline with stacked LSTM, seq2seq with GRU, and seq2seq with LSTM.

2. RELATED WORKS

For many years, DL has been a major improvement in ML research, solving high-dimensional data problems. [15]. DL is used in many domains of science, business, and government. The most popular deep learning methods which are used for predicting indoor climate problems are Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), a simplified LSTM.

There is also some research which used deep learning to predict indoor climate problems. The closest work to our research is the research done by Gunawan et al. [16]. They developed four deep learning models for predicting indoor temperature and humidity, such as LSTM, GRU, Transformer, and Transformer with learnable positional encoding. Their datasets contained indoor temperature, indoor humidity, and 2 lighting variables in 3 different places. Their results show that the GRU model was superior for all humidity predictions, and the LSTM model was superior for 2 of 3 temperature predictions.

Another related research was about predicting indoor climate parameters such as temperature, humidity, and carbon dioxide concentration inside a greenhouse for tomato plants by Ali and Hassanein [17]. They used the LSTM model as their prediction model. Similarly to Ali and Hassanein, Jung et al. predicted indoor climate such as temperature, humidity, and carbon dioxide concentration inside a greenhouse, but they compared three different models such as ANN-Backpropagation, Nonlinear Autoregressive Exogenous model (NARX), and LSTM with datasets obtained from Davis wireless vantage Pro2 (Davis instruments, California, USA) weather station, and HMP 35 Probe (Vasaila, Helsinki, Finland). [18]. The result concluded that LSTM was superior to ANN-Backpropagation and NARX.

Another indoor climate prediction research was done by Liu et al [19]. Their research implemented time sliding window to their LSTM model for learning the change of environment climate over short a period of time. Their datasets were tomato, cucumber, and spicy greenhouse with indoor temperature and humidity, light intensity, carbon dioxide concentration, soil temperature, and soil humidity. Their modified LSTM outperformed the GRU model.

Elhariri and Taie conducted a similar study to SDD in which they experimented with Heating, Ventilation, and Air Conditioning (HVAC), an indoor system similar to SDD [20]. They compared LSTM and GRU models to predict the future microclimate inside smart buildings by using UCI Machine Learning Repository SML2010 datasets containing indoor temperature and humidity, carbon dioxide concentration, and outdoor temperature and humidity. The result showed that the GRU model was the best in their case.

The research that is closest to ours, which implemented the seq2seq architecture as a model time series prediction, was done by Fang et al. [21]. They predicted an indoor climate inside the GreEn-ER building in the center of Grenoble, France, with the datasets containing indoor temperature and carbon dioxide. They proposed 3 seq2seq models, such as LSTM-Dense, LSTM-LSTM, and LSTM-Dense-LSTM, which outperform the LSTM and GRU baseline models.

Inspired by the succession of the seq2seq architecture by Fang et al., this research implemented LSTM and GRU in our seq2seq models to be compared with our proposed adapted baseline with stacked LSTM and stacked GRU models.

SDD can accomplish many tasks by accurately predicting future indoor climates, such as regulating indoor climatic conditions, attaining ideal agricultural product drying conditions, and minimizing energy use[22]. Inspired by Fang et al. and Gunawan et al., this study proposed 4 models to be compared, such as the adapted GRU-based sequence baseline model, the adapted LSTM-based sequence baseline model, seq2seq GRU, and seq2seq LSTM [16] [21].

3. DATA AND METHODOLOGY

3.1. Datasets

The dataset which was used in this experiment was generated from a SDD facility in Sumedang, a town in Western Java, Indonesia. The facility can be seen in Figure 1.

FIGURE 1. SDD Facility in Sumedang, Indonesia.

The datasets were generated from two indoor sensors and an outdoor sensor, containing temperature and humidity data for each sensor over 12 days of recording. The dataset on the SDD can be depicted in Figure 2, with temperature represented by a blue line and humidity represented by an orange line.

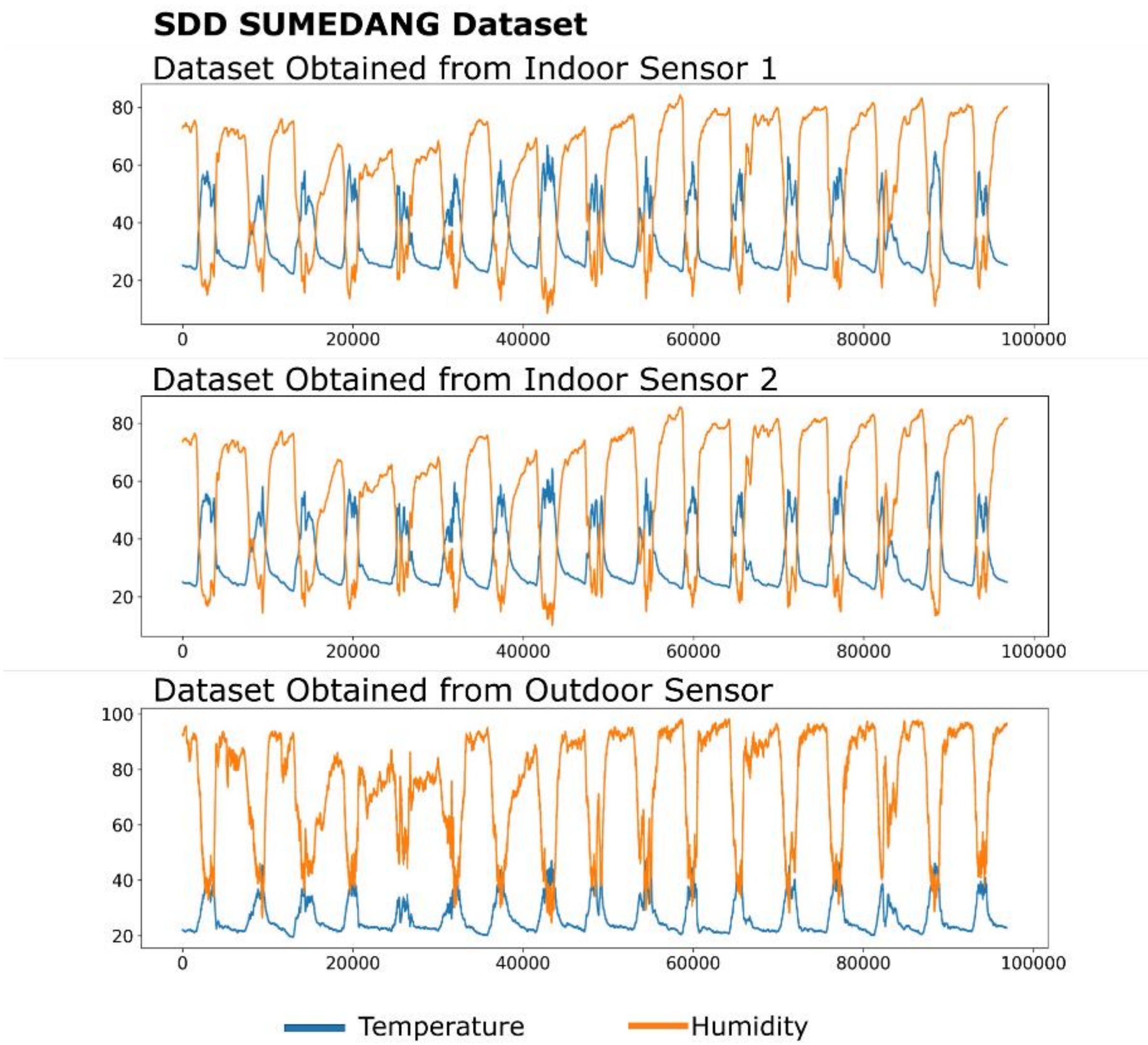

FIGURE 2. Datasets Obtained from Sensor.

This research only addressed indoor temperature and indoor humidity, even though all models forecasted all six features by using all six features as input and output. Because the two primary factors that affect SDD, which need to be monitored and controlled, are indoor temperature and indoor humidity. So, the results of the outdoor temperature and the outdoor humidity were ignored.

3.2. Long Short-Term Memory

Due to its capacity for memorizing temporal information over a large number of timesteps, Long Short-Term Memory (LSTM) is frequently employed in classification and regression tasks involving sequential data [23]. LSTM is composed of three gates, which are the input gate, output gate, and forget gate. An LSTM was designed well to handle time series predictions and is also a solution for problems which require temporal memory [24].

$$i^{(t)} = \sigma(W_i x^{(t)} + R_i y^{(t-1)} + p_i \odot c^{(t-1)} + b_i) \tag{1}$$

The input gate in equation (1) is denoted as $i^{(t)}$ where $x^{(t)}$, $c^{(t-1)}$, and $y^{(t-1)}$ are the representations for the input data, last iteration output data, and last iteration cell value respectively with W_i, R_i, and p_i as weight values. The bias vector of input gate in LSTM is indicated by the symbol of b_i. The symbol of σ denotes the sigmoid activation function.

$$f^{(t)} = \sigma(W_f x^{(t)} + R_f y^{(t-1)} + p_f \odot c^{(t-1)} + b_f) \tag{2}$$

The forget gate in equation (2) is denoted as $f^{(t)}$ which eliminates the information from previous cell state where W_f, R_f, and p_f symbolize the weight values for input data, last iteration output data, and last iteration cell value respectively. The bias vector of forget gate in LSTM is symbolized as b_f.

$$c^t = z^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)} \tag{3}$$

The cell value in equation (3) is denoted as c^t where $z^{(t)}$ is the block input.

$$o^{(t)} = \sigma(W_o x^{(t)} + R_o y^{(t-1)} + p_o \odot c^{(t)} + b_o) \tag{4}$$

The output gate in equation (4) is denoted as $o^{(t)}$ where W_o, R_o, and p_o are the weight values for input data, last iteration output data, and last iteration cell value respectively.

$$y^{(t)} = g(c^{(t)}) \odot o^{(t)} \tag{5}$$

The block output of LSTM in equation (5) is denoted as $y^{(t)}$ which combine current cell value and the output gate in LSTM where $g(x)$ is hyperbolic tangent function.

3.3. Gated Recurrent Unit

Gated Recurrent Unit (GRU) is a simplified LSTM with 1 less gate [25]. GRU mostly outperforms LSTM in many cases [26]. If LSTM has 3 gates, which are input gate, forget gate, and output gate, GRU has 2 gates, which are reset gate symbolized as r_t and update gate symbolized as z_t [27].

$$\widetilde{s_t} = \phi_{tanh}(W_s(r_t \odot s_{t-1}) + U_s x_t + b_s) \tag{6}$$

$$s_t = (1 - z_t) \odot s_{t-1} + z_t \odot \widetilde{s_t} \tag{7}$$

$$r_t = \sigma_{sig}(W_r s_{t-1} + U_r x_t + b_r) \tag{8}$$

$$z_t = \sigma_{sig}(W_z s_{t-1} + U_z x_t + b_z) \tag{9}$$

where: $\odot$ is element-wise multiplier; W_s, W_r, and W_z are weight value; x_t is input data; $\widetilde{s_t}$ is candidate state; s_t is output; b_s, b_r, b_z are constants; σ_{sig} and ϕ_{tanh} are sigmoid and tanh activation function.

3.4. Prediction Model Architectures

3.4.1 Adapted Baseline Sequence Models

This research modified the models implemented by Gunawan et al. to handle sequence inputs and sequence outputs [16]. The adapted baseline sequence models consisted of the two most popular RNN layers, which are LSTM and GRU, which can be seen in Figure 3.

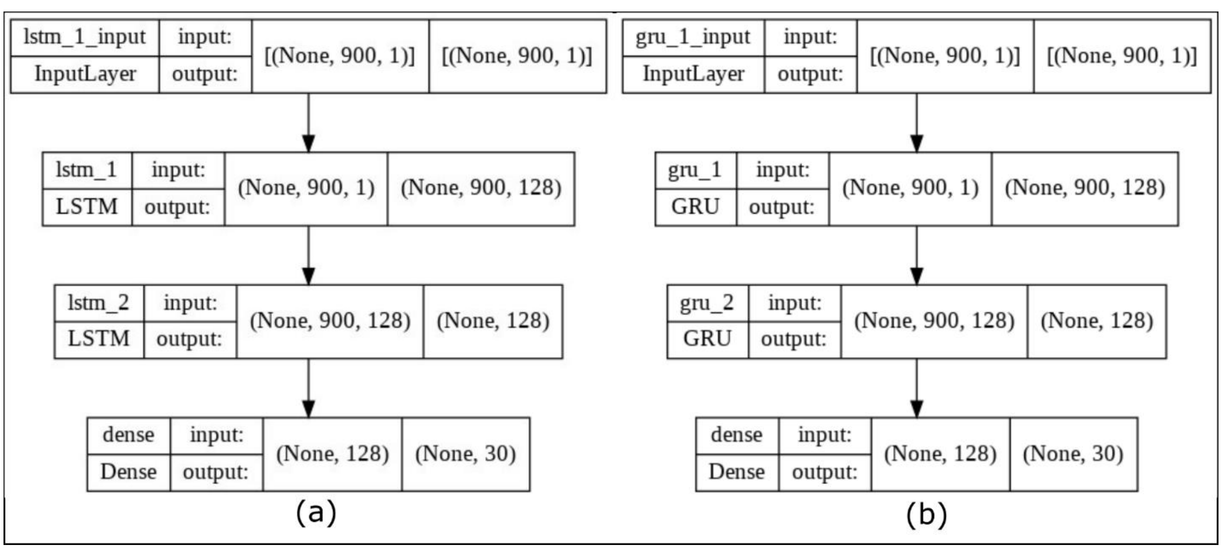

FIGURE 3. Adapted Baseline Architecture with Stacked LSTM (a) and Stacked GRU (b).

Since the datasets in this research were processed as 3-dimensional (3D) data because of the sliding window process, the 3D data input as (i, k, j) of adapted baseline sequence models were reshaped to 2D data $(i, k \times j)$ with i representing the amount of sliding window process, k representing timestep, and j representing the number of features. The reshaping process is illustrated in Figure 4. Because the output of adapted baseline models is 2D data $(i, k \times j)$, the output needed to be reshaped back to 3D data (i, k, j).

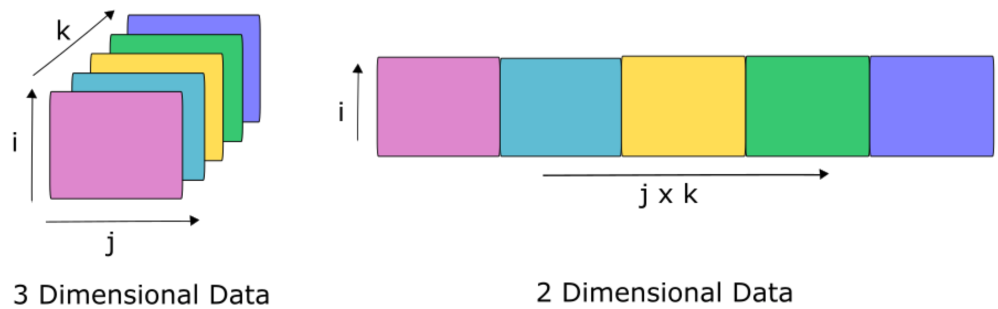

FIGURE 4. Reshaping Illustration.

The hyperparameters of the adapted baseline models followed the baseline settings by Gunawan et al., such as implementing 128 neurons on both LSTM and GRU layers, 64 batch size, 0.001 learning rate, and 100 epochs with the Adam optimization algorithm [16].

3.4.2 Sequence-to-sequence (Seq2seq) or Encoder-Decoder Models

The encoder-decoder or sequence-to-sequence (seq2seq) is also part of deep learning, which originated from machine translation problems, where at the beginning of its appearance, the seq2seq architecture could empirically perform well for translation tasks from English to French [28]. Seq2seq consists of two Recurrent Neural Networks (RNN) which act as the encoder and decoder. The Seq2seq architecture is mostly used for language processing models [29] and has rarely been used for indoor climate forecasting [21]. The Seq2seq model also performed well in predicting time-series data, like predicting Beijing PM25 datasets, energy consumption in Sceaux, highway traffic in the UK, Italian air quality, and California traffic with PeMS-Bays datasets [30].

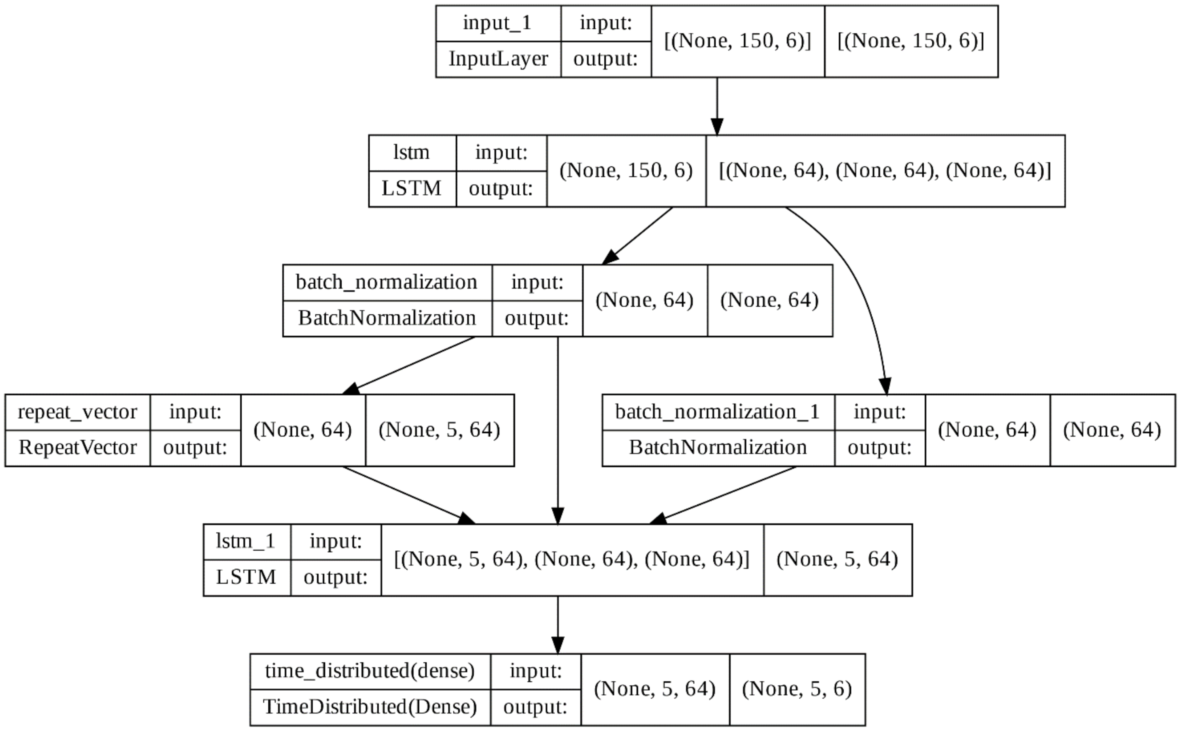

FIGURE 5. Architecture of LSTM Seq2seq.

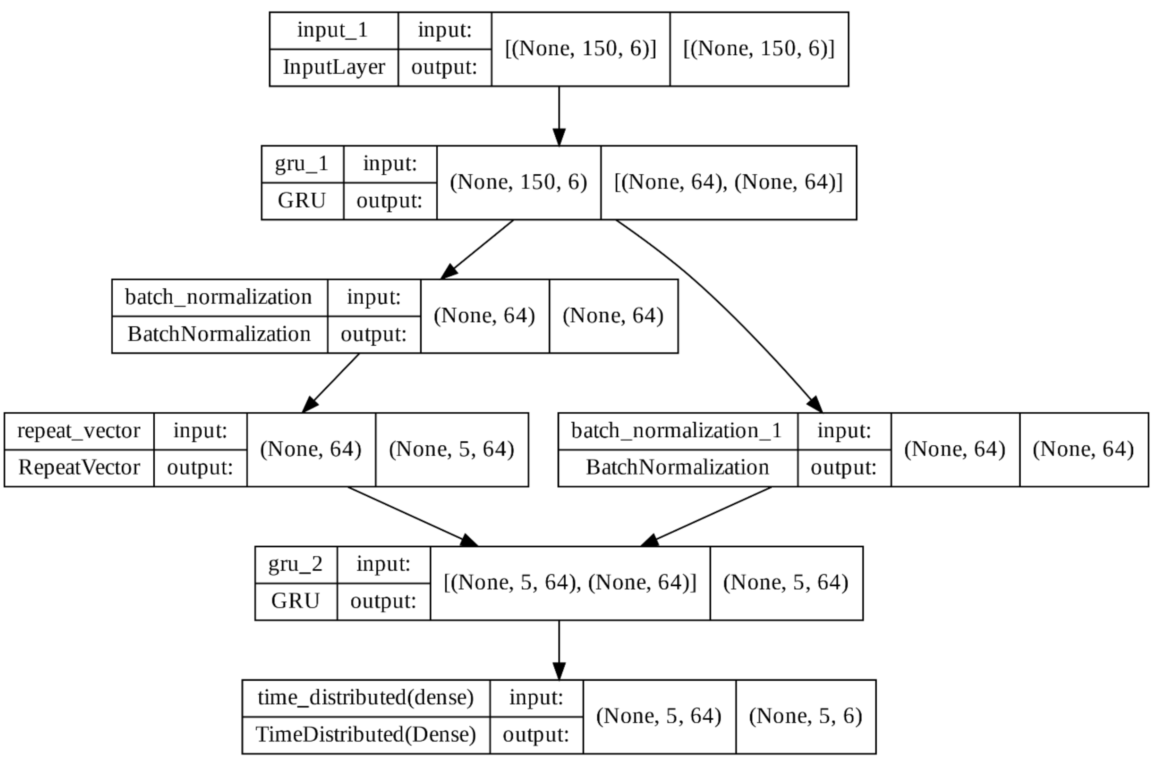

FIGURE 6. Architecture of GRU Seq2seq.

This research implemented two simple seq2seq architectures with RNN layers such as LSTM and GRU used in the encoder and decoder layers, which can be seen in Figure 5 and 6. Both seq2seq architectures implemented batch normalization between encoder and decoder. Batch normalization can improve the accuracy and generalization [12], and accelerate the training process, which makes it one of the favorite techniques in deep learning [31], because seq2seq is more complex than our adapted baseline models.

To obtain suitable hyperparameter settings for the seq2seq model, this research observed and understood the early training with a short run of 10 epochs by doing random search, because by doing that, it could be a clue for suitable model settings without consuming time and expensive computational resources [32]. The result of random search observation showed that 64 neurons for each GRU and LSTM layer and a learning rate with 0.00001 provided the best results. Then this research equated to 64 batch size, 100 epochs, and implemented Adam optimization algorithm. Adam was chosen because it has advantages over other adaptive learning rate optimization algorithms, including the ability to handle non-stationary objectives like RMSProp and manage sparse gradients like AdaGrad [33].

3.5. Pearson Correlation

$$r_{xy} = \frac{\sum(x_i - \bar{x}) \sum(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} \tag{10}$$

To find out the correlation between each parameter, Pearson Correlation Coefficient (PCC), which is denoted as r_{xy} was implemented where x and y are the compared parameters, $\bar{x}$ and $\bar{y}$ are the mean value of x and y respectively [34]. The PCC results will be in the range [-1,1] where $r_{xy} = -1$ means that the correlation is extremely negative and $r_{xy} = 1$ is conversely [35].

3.6. Performance Metrics

The most commonly used performance metrics which are implemented in regression analysis cases in machine learning studies are Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE) [36]. In fact, each error measurement has different disadvantages that can lead to inaccurate evaluation of forecasting results, which makes it not recommended to only use one measurement [37]. This research aimed to forecast indoor temperature and humidity in the future, which made MAE and RMSE an ideal choice for collecting error information in the model. This research also implemented the coefficient of determination (R^2) because of its potential to compare ground truth elements with predicted data considering its distribution [36].

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|X_i - Y_i| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2} \tag{12}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2} \tag{13}$$

In MAE, RMSE and R^2 equations, X_i is the predicted value at i^{th}, Y_i is the ground truth data at i^{th} and $\bar{Y}$ represent mean of ground truth data. Both MAE and RMSE results must be in range [0,∞) with the best value is closer to 0, meanwhile in R^2 result will be in range (-∞, 1] with the best value is closer to 1.

R^2 value describes the proportion of variance in a variable which is affected by another variable [38]. R^2 can be categorized as strong when $R^2 \geq 0.75$ and weak when $R^2 \leq 0.25$ [39]. Meanwhile between both strong and weak, there is moderate R^2 value.

4. EXPERIMENTS

4.1. Experimental Environments

In this research, TensorFlow Python version 2.8.2 library and Keras version 2.80 library were used to train the model. The operating system was Ubuntu 20.04. The graphics card, used in this research was NVIDIA Quadro RTX 8000 with 127GB of RAM.

4.2. Features Correlation in Dataset

Figure 7 depicts that each PCC value among all parameters with temp1, hum1, temp2, and hum2 are indoor parameters, whereas temp3 and hum3 are outdoor parameters. The PCC result shows that between temperature parameters there are extremely strong positive values with 0.97 to 1, and so do the humidity parameters. Meanwhile, the correlations between temperature and humidity parameters show extremely negative PCC values with less than -0.92.

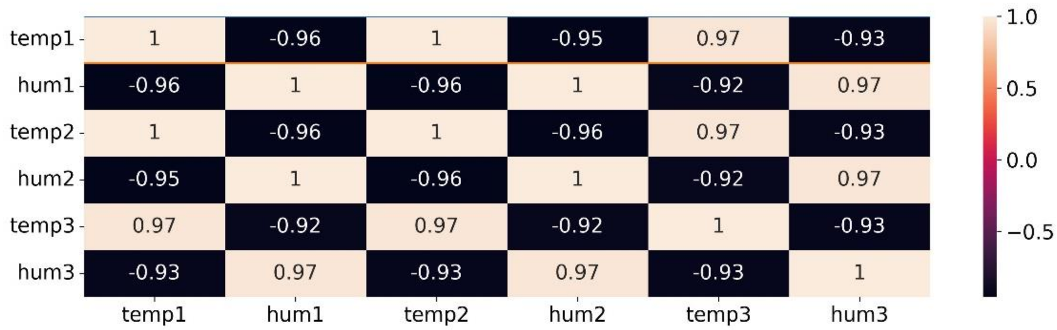

FIGURE 7. PCC Values among All Dataset Parameters.

4.3. Preprocessing Dataset

The datasets, contained enormous datasets with appropriate time series data, which made them suitable for deep learning models [40] [41], as shown in Figure 2. This study divided the dataset into two parts: training data and test data, with a percentage of 80% and 20%, respectively. The model presented in this study consumed training data in the training step, with 80% of the training data being used to train the model and 20% of the training data being used to validate the model.

Because extremely high or low data values can trigger the models to overfit, data standardization was used in this experiment research to assist the models in learning the data [42].

$$ZN(x) = \frac{x - \mu(x)}{\sigma(x)} \tag{14}$$

In this study, Z-score standardization, designated as $ZN(x)$ with mean (μ) and standard deviation as (σ) and, is used. Figures 8 show the outcomes of applying Z-score standardization to our datasets.

Figure 8 show how standardized datasets were divided into two parts: training and testing. Relative humidity data in orange and temperature data in blue were used to train models, while relative humidity data in red and temperature data in green were used to test models.

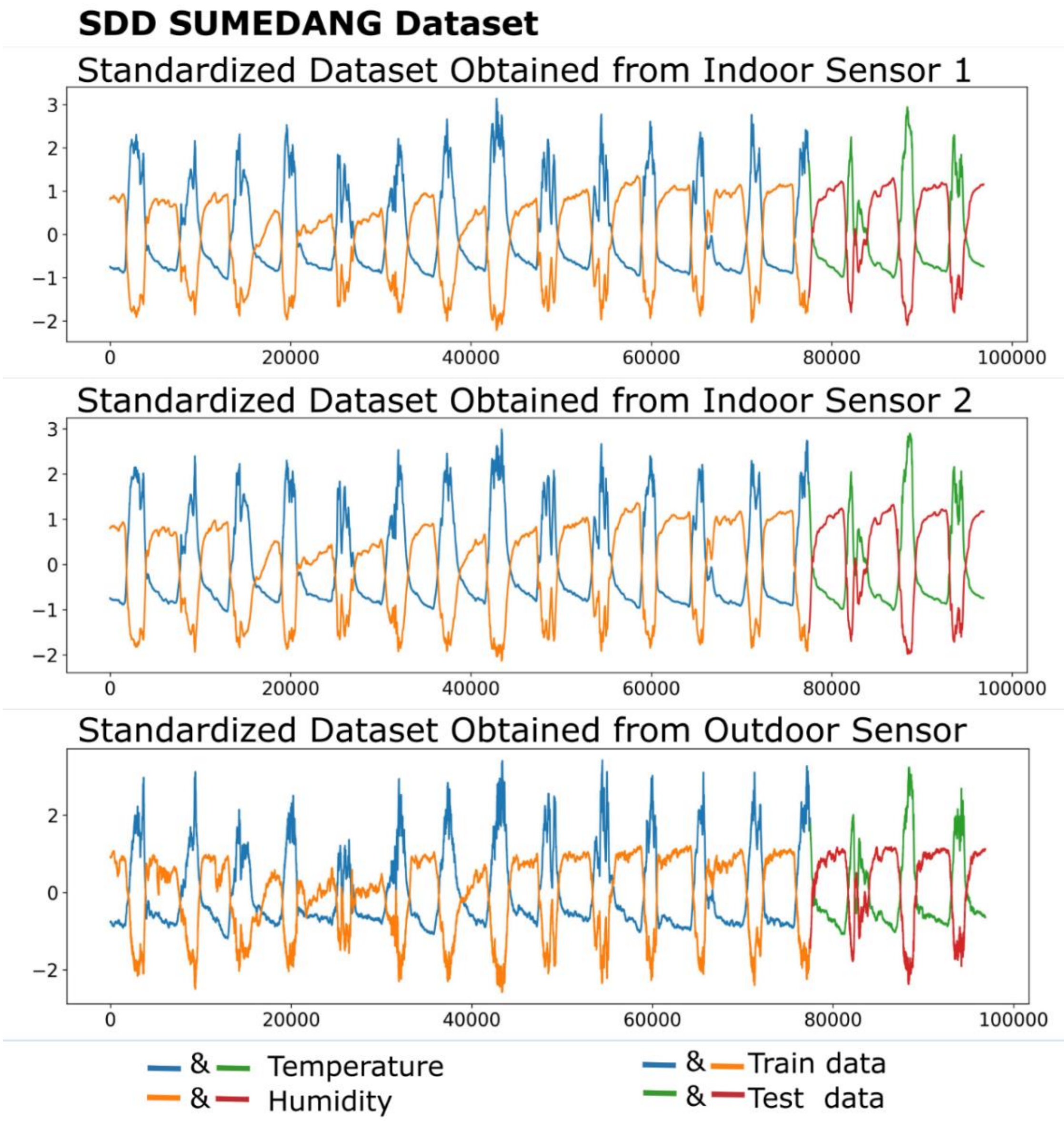

FIGURE 8. Data Standardization Results.

Accordingly, the data needed to be transformed from 2D data illustrated as (a, b), where a represents the amount of data and b represents a data feature such as indoor temperature 1, indoor humidity 1, indoor temperature 2, indoor humidity 2, outdoor temperature, and outdoor humidity, into 3D data illustrated as (c, d, b), where c represents the number of smaller pieces of data partition and d represents the timesteps number of input or output data. This research aims to predict data in the future 5 timesteps based on 150 previous data, because five timesteps of predicted data should be enough to support SDD operational. Figure 9 illustrates the sliding process.

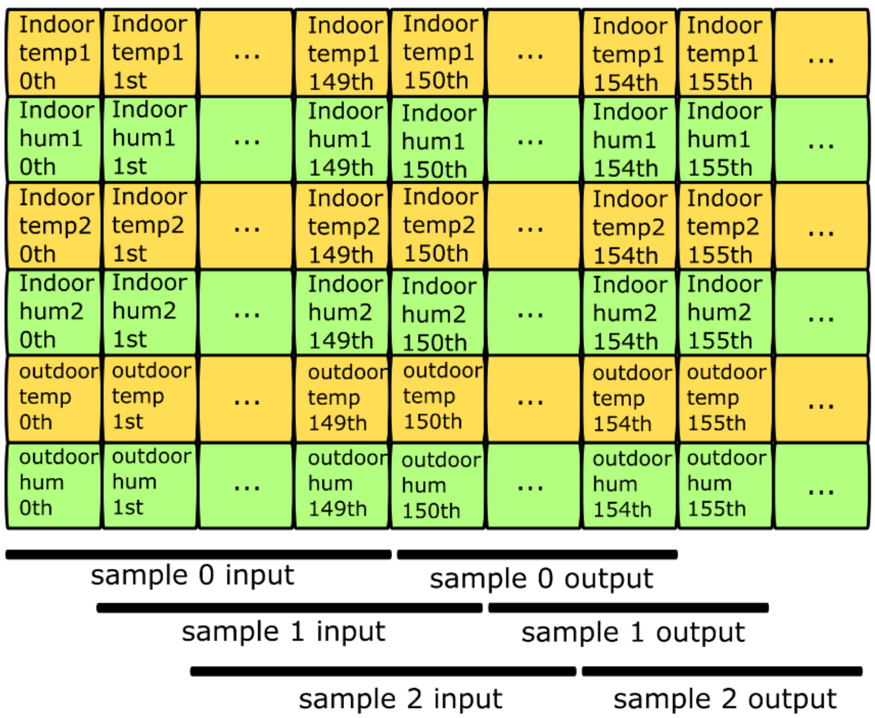

FIGURE 9. Input and Output Data after Sliding Window Process.

4.4. Model Training

Figure 10 shows that both adapted LSTM and GRU baseline models were better in learning datasets containing extremely strong positive and negative PCC. Those adapted baseline sequence models can reach a loss value below 0.01 in MAE in both the training and validation processes. Meanwhile, both seq2seq LSTM and GRU can only reach a loss value of around 0.05 in MAE in both training and validation. The training and validation processes conclude that seq2seq models were too complicated to learn the datasets containing only extremely strong PCC and the adapted baseline models were simple enough and suitable to learn these datasets.

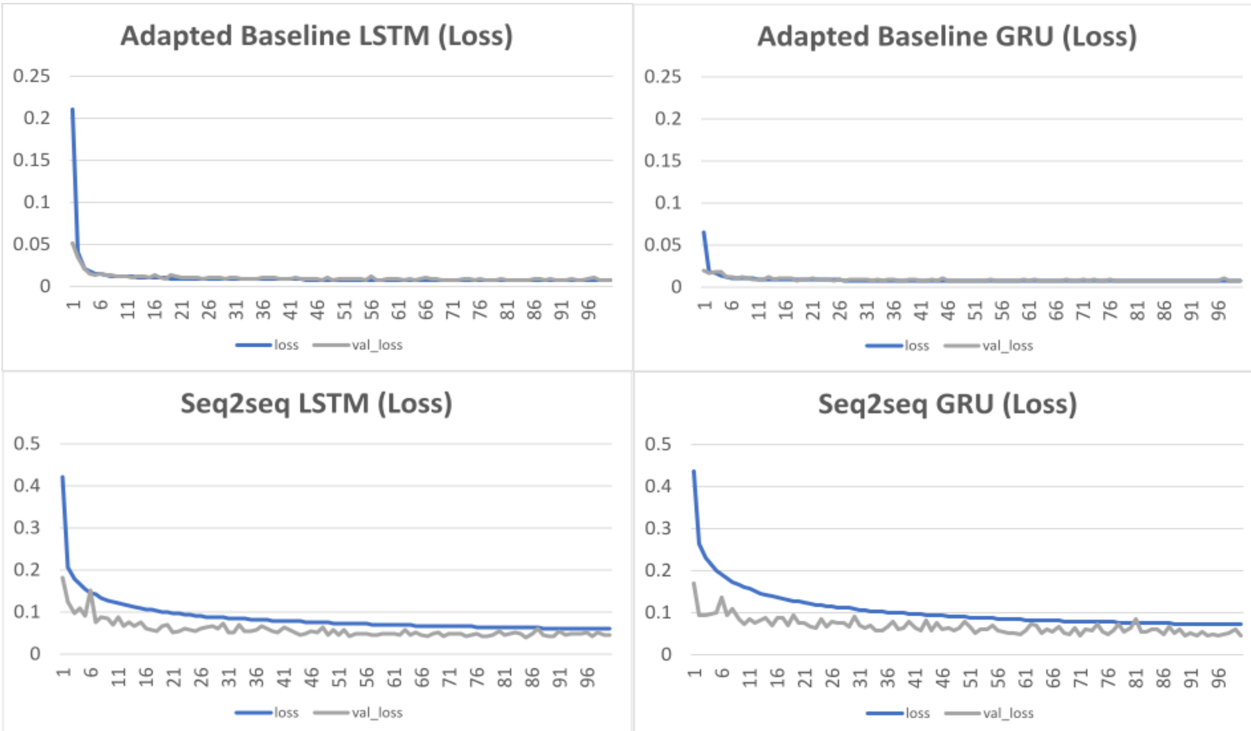

FIGURE 10. Train and Validation Loss Plot in MAE of All Models.

4.5. Model Results and Comparison

This research compared all prediction models by using testing data, which was untrained data, or a fifth of the original datasets. Untrained data, depicted in Figure 8, as red and green lines on the line chart, was fed into the model, yielding predicted data (X_i). Then this study compared both predicted data (X_i) and ground truth data (Y_i) with MAE, RMSE, and R^2 as performance metrics.

Since the prediction result was standardized data with Z-score, the results needed to be converted back to real ranges. The prediction model results (X_i) and ground truth data (Y_i) contained 6 features, including indoor temperature and humidity 1, indoor temperature and humidity 2, and outdoor temperature and humidity with 5 timesteps. Both sets of data were compared with MAE, RMSE, and R^2 as performance metrics.

A quick glance at Tables 1 and 2 shows that indoor temperature prediction testing using the adapted GRU baseline produced the best results in MAE, RMSE, and R^2. The results show significant differences between adapted sequence baseline models and seq2seq models where the adapted LSTM baseline outperformed the seq2seq LSTM model by an average difference in both indoor temperature prediction of 0.3473 in MAE, and the adapted GRU baseline outperformed the seq2seq GRU model by an average difference in both indoor temperature prediction of 0.3766 in MAE.

TABLE 1. Indoor Temperature 1 Testing Results.

Time	Error Metrics	Models			
		Adapted Sequence Baseline Models		Seq2seq Models	
		Adapted LSTM Baseline	Adapted GRU Baseline	Seq2seq LSTM	Seq2seq GRU
Overall	MAE	0.064542	**0.062371**	0.418707	0.422684
	RMSE	0.093637	**0.086790**	0.626587	0.757507
	R^2	0.999916	**0.999927**	0.996169	0.993926
t+1	MAE	0.052800	**0.044020**	0.398213	0.335650
	RMSE	0.068675	**0.057263**	0.585514	0.651003
	R^2	0.999955	**0.999968**	0.996666	0.995626
t+2	MAE	0.052707	**0.048344**	0.392174	0.420372
	RMSE	0.076581	**0.061438**	0.595027	0.748065
	R^2	0.999944	**0.999964**	0.996542	0.994048
t+3	MAE	0.061645	**0.068161**	0.415084	0.455739
	RMSE	0.088463	**0.086884**	0.618344	0.796650
	R^2	0.999925	**0.999927**	0.996287	0.993199
t+4	MAE	0.072228	**0.058176**	0.437910	0.454064
	RMSE	0.102411	**0.090305**	0.650243	0.799247
	R^2	0.999899	**0.999921**	0.995878	0.993193
t+5	MAE	**0.083332**	0.093151	0.450155	0.447593
	RMSE	0.122248	**0.122087**	0.678977	0.782450
	R^2	**0.999856**	0.999855	0.995474	0.993562

TABLE 2. Indoor Temperature 2 Testing Results.

Time	Error Metrics	Models			
		Adapted Sequence Baseline Models		Seq2seq Models	
		Adapted LSTM Baseline	Adapted GRU Baseline	Seq2seq LSTM	Seq2seq GRU
Overall	MAE	0.072104	**0.057594**	0.412551	0.450679
	RMSE	0.114922	**0.080567**	0.624098	0.919523
	R^2	0.999870	**0.999936**	0.996158	0.990810
t+1	MAE	0.056056	**0.035350**	0.432378	0.440503
	RMSE	0.088789	**0.047310**	0.605891	0.870594
	R^2	0.999923	**0.999978**	0.996466	0.991877
t+2	MAE	0.059912	**0.048755**	0.364134	0.456120
	RMSE	0.087589	**0.061664**	0.565115	0.946860
	R^2	0.999925	**0.999963**	0.996854	0.990186
t+3	MAE	0.075596	**0.059553**	0.403863	0.450419
	RMSE	0.114746	**0.077221**	0.608699	0.952605
	R^2	0.999871	**0.999941**	0.996352	0.990020
t+4	MAE	0.081308	**0.070819**	0.423658	0.450090
	RMSE	0.125917	**0.092996**	0.647581	0.928949
	R^2	**0.991026**	0.990744	0.995839	0.990592
t+5	MAE	0.087647	**0.073492**	0.438720	0.456265
	RMSE	0.146487	**0.108643**	0.686407	0.895954
	R^2	0.999790	**0.999884**	0.995280	0.991377

The results of indoor humidity prediction testing based on Tables 3 and 4 show a similar trend with indoor temperature prediction testing results that the seq2seq models outperformed the adapted sequence baseline models, where the adapted LSTM baseline model performed better than the seq2seq LSTM model with an average MAE difference in both indoor humidity prediction of 0.9127, and the adapted GRU baseline model performed better than the seq2seq GRU model with an average MAE difference in both indoor humidity prediction of 0.6046.

TABLE 3. Indoor Humidity 1 Testing Results.

Time	Error Metrics	Models			
		Adapted Sequence Baseline Models		Seq2seq Models	
		Adapted LSTM Baseline	Adapted GRU Baseline	Seq2seq LSTM	Seq2seq GRU
Overall	MAE	**0.087518**	0.097254	1.124548	0.724641
	RMSE	**0.125062**	0.138359	1.356994	1.059923
	R^2	**0.999965**	0.999956	0.996109	0.997472
t+1	MAE	0.076289	**0.071344**	0.977144	0.626122
	RMSE	0.103721	**0.095238**	1.221450	0.905692
	R^2	0.999976	**0.999979**	0.996823	0.998169
t+2	MAE	**0.080836**	0.089261	1.109321	0.703736
	RMSE	**0.108182**	0.119016	1.312224	0.998131
	R^2	**0.999974**	0.999968	0.996356	0.997757
t+3	MAE	**0.086753**	0.091218	1.214375	0.718366
	RMSE	**0.119617**	0.125580	1.428900	1.050451
	R^2	**0.999968**	0.999964	0.995716	0.997503
t+4	MAE	**0.092664**	0.107779	1.199042	0.755174
	RMSE	**0.134137**	0.154716	1.431309	1.116251
	R^2	**0.999959**	0.999945	0.995691	0.997185
t+5	MAE	**0.101051**	0.126666	1.122856	0.819808
	RMSE	**0.153102**	0.181035	1.379455	1.204648
	R^2	**0.999947**	0.999925	0.995961	0.996747

In Table 3, an interesting thing happened. The adapted LSTM model outperformed all models in predicting indoor humidity 1, while the adapted GRU baseline outperformed all models in predicting indoor temperature 1, temperature 2, and humidity 2.

TABLE 4. Indoor Humidity 2 Testing Results.

Time	Error Metrics	Models			
		Adapted Sequence Baseline Models		Seq2seq Models	
		Adapted LSTM Baseline	Adapted GRU Baseline	Seq2seq LSTM	Seq2seq GRU
Overall	MAE	0.098777	**0.095241**	0.887168	0.677099
	RMSE	0.169456	**0.167658**	1.157673	0.950485
	R^2	0.999938	**0.999939**	0.997292	0.998072
t+1	MAE	0.068244	**0.061280**	0.917527	0.544288
	RMSE	0.113080	**0.111591**	1.162305	0.735596
	R^2	0.999972	**0.999973**	0.997284	0.998856
t+2	MAE	0.099880	**0.079493**	0.849962	0.682674
	RMSE	0.154163	**0.143332**	1.124332	0.898333
	R^2	0.999949	**0.999955**	0.997445	0.998296
t+3	MAE	**0.089917**	0.094333	0.890735	0.678909
	RMSE	**0.162347**	0.165702	1.153160	0.938349
	R^2	**0.999943**	0.999940	0.997315	0.998118
t+4	MAE	**0.104501**	0.105365	0.898606	0.711925
	RMSE	**0.181975**	0.183828	1.171240	1.018065
	R^2	**0.999928**	0.999926	0.997226	0.997775
t+5	MAE	**0.131342**	0.135734	0.879009	0.767701
	RMSE	0.218064	**0.215172**	1.176594	1.118953
	R^2	0.999897	**0.999899**	0.997191	0.997317

The results in Tables 1, 2, 3, and 4 show that in overall prediction, both the adapted baseline with LSTM and GRU outperformed the seq2seq model with LSTM and GRU, with the number in bold representing the best result. In predicting indoor temperature, the adapted baseline model with GRU was the best. Meanwhile, in predicting indoor humidity, both the adapted baseline model with GRU and LSTM were comparable. The Seq2seq model, a deep learning model which is popular in natural language processing [43], is more complex than the adapted baseline model to handle a dataset containing extremely strong positive PCC values between the same temperature parameters or humidity parameters and extremely negative PCC values between temperature parameters and humidity parameters. A quick glance at all the testing result tables shows that seq2seq models produced ten times higher error than adapted sequence baseline models, but all models were still good at predicting indoor climate with an average of $MAE \leq 0.5$ for indoor temperature prediction and $MAE \leq 1.2$ for indoor humidity prediction. With coefficient of determination, all of the models can be categorized as strong in with $R^2 \geq 0.99$ [39].

The dataset used in this research contained a large amount of real-time sensor data, which had the possibility of containing noise data [44]. So, for future work, the next research will implement Kalman filtering to correct the data from noise in order to increase the accuracy of all models.

5. Conclusion

The results show that in processing the dataset, which contained only extremely strong positive or extremely strong negative PCC values between each other parameter, both our adapted baseline models outperformed both our seq2seq models. All models were good at predicting indoor temperature and humidity because they had a relatively small error number in MAE and RMSE. The coefficient determination values of all models were also categorized as strong, with $R^2 \geq 0.99$.

Based on this research, the curiosity arose because seq2seq models still have the potential to be improved, such as by implementing attention layers. In future research, there is a plan to improve seq2seq architectures by adding an attention layer and stacking some RNN layers inside both the encoder and decoder layers and improving the case to be a more complex problem, such as increasing the number of timesteps in both input and output models. Because of the dataset containing a large amount of time series data captured by sensors inside SDD, there will be a future study on reducing noise from the dataset by using a filtering technique such as Kalman filtering.

ACKNOWLEDGEMENTS

The experiments in this study used a computer with NVIDIA Quadro RTX 8000 with 127GB of RAM facilitated by NVIDIA-BINUS Artificial Intelligence Research and Development Center (NVIDIA-AIRDC). The authors are grateful to the Electrical Engineering Department at Trisakti University for helping to provide the dataset from SDD facility in Sumedang, Indonesia.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] A.S. Budiman, F. Gunawan, E. Djuana, et al. Smart dome 4.0: Low-cost, independent, automated energy system for agricultural purposes enabled by machine learning, J. Phys.: Conf. Ser. 2224 (2022), 012118. https://doi.org/10.1088/1742-6596/2224/1/012118.

[2] G. Srinivasan, P. Muthukumar, A review on solar greenhouse dryer: Design, thermal modelling, energy, economic and environmental aspects, Solar Energy. 229 (2021), 3–21. https://doi.org/10.1016/j.solener.2021.04.058.

[3] F.E. Gunawan, A.S. Budiman, B. Pardamean, et al. Design and energy assessment of a new hybrid solar drying dome - Enabling Low-Cost, Independent and Smart Solar Dryer for Indonesia Agriculture 4.0, IOP Conf. Ser.: Earth Environ. Sci. 998 (2022), 012052. https://doi.org/10.1088/1755-1315/998/1/012052.

[4] R.E. Caraka, R.C. Chen, S.A. Bakar, Employing best input SVR robust lost function with nature-inspired metaheuristics in wind speed energy forecasting, IAENG Int. J. Comput. Sci. 47 (2020), 572–584.

[5] R.E. Caraka, B.D. Supatmanto, M. Tahmid, et al. Rainfall forecasting using PSPline and rice production with ocean-atmosphere interaction, IOP Conf. Ser.: Earth Environ. Sci. 195 (2018) 012064. https://doi.org/10.1088/1755-1315/195/1/012064.

[6] D.N.N. Putri, D.P. Adji, Stevanus, et al. Power system design for solar dryer dome in agriculture, in: 3rd International Conference on Sustainable Engineering and Creative Computing (ICSECC) 2021, Cikarang, Indonesia.

[7] R.E. Caraka, R.C. Chen, H. Yasin, et al. Hybrid vector autoregression feedforward neural network with genetic algorithm model for forecasting space-time pollution data, Indonesian J. Sci. Technol. 6 (2021), 243–266. https://doi.org/10.17509/ijost.v6i1.32732.

[8] F.Q. Lauzon, An introduction to deep learning, in: 2012 11th International Conference on Information Science, Signal Processing and Their Applications (ISSPA), IEEE, Montreal, QC, Canada, 2012: pp. 1438–1439. https://doi.org/10.1109/ISSPA.2012.6310529.

[9] H. Prabowo, A.A. Hidayat, T.W. Cenggoro, et al. Aggregating time series and tabular data in deep learning model for university students' GPA prediction, IEEE Access. 9 (2021), 87370–87377. https://doi.org/10.1109/access.2021.3088152.

[10] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Physica D: Nonlinear Phenomena. 404 (2020), 132306. https://doi.org/10.1016/j.physd.2019.132306.

[11] A. Budiarto, R. Rahutomo, H.N. Putra, et al. Unsupervised news topic modelling with Doc2Vec and spherical clustering, Procedia Computer Sci. 179 (2021), 40–46. https://doi.org/10.1016/j.procs.2020.12.007.

[12] Z. Shen, Y. Zhang, J. Lu, et al. A novel time series forecasting model with deep learning, Neurocomputing. 396 (2020), 302–313. https://doi.org/10.1016/j.neucom.2018.12.084.

[13] A. Shewalkar, D. Nyavanandi, S.A. Ludwig, Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU, J. Artif. Intell. Soft Comput. Res. 9 (2019) 235–245. https://doi.org/10.2478/jaiscr-2019-0006.

[14] S. Hwang, G. Jeon, J. Jeong, et al. A novel time series based Seq2Seq model for temperature prediction in firing furnace process, Procedia Computer Sci. 155 (2019), 19–26. https://doi.org/10.1016/j.procs.2019.08.007.

[15] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature. 521 (2015), 436–444. https://doi.org/10.1038/nature14539.

[16] F. E. Gunawan, A.S. Budiman, B. Pardamean et al. Multivariate time-series deep learning for joint prediction of temperature and relative humidity in a closed space, in: 2021 International Conference on Computer Science and Computational Intelligence, 2021.

[17] A. Ali, H.S. Hassanein, Wireless sensor network and deep learning for prediction greenhouse environments, in: 2019 International Conference on Smart Applications, Communications and Networking (SmartNets), IEEE,

Sharm El Sheik, Egypt, 2019: pp. 1–5. https://doi.org/10.1109/SmartNets48225.2019.9069766.

[18] D.H. Jung, H.S. Kim, C. Jhin, et al. Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse, Computers Electron. Agric. 173 (2020), 105402. https://doi.org/10.1016/j.compag.2020.105402.

[19] Y. Liu, D. Li, S. Wan, et al. A long short - term memory - based model for greenhouse climate prediction, Int. J. Intell. Syst. 37 (2021), 135 - 151. https://doi.org/10.1002/int.22620.

[20] E. Elhariri, S.A. Taie, H-Ahead multivariate microclimate forecasting system based on deep learning, in: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), IEEE, Aswan, Egypt, 2019: pp. 168–173. https://doi.org/10.1109/ITCE.2019.8646540.

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

[22] R.E. Caraka, R.C. Chen, T. Toharudin, et al. Evaluation performance of SVR genetic algorithm and hybrid PSO in rainfall forecasting, ICIC Express Lett. Part B. Appl. 11 (2020), 631-639. https://doi.org/10.24507/icicelb.11.07.631.

[23] Z. Han, J. Zhao, H. Leung, et al. A review of deep learning models for time series prediction, IEEE Sensors J. 21 (2021), 7833–7848. https://doi.org/10.1109/jsen.2019.2923982.

[24] G. Van Houdt, C. Mosquera, G. Nápoles, A review on the long short-term memory model, Artif. Intell. Rev. 53 (2020), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1.

[25] J. Chung, C. Gulcehre, K. Cho, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling, (2014). http://arxiv.org/abs/1412.3555.

[26] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE, Boston, MA, 2017: pp. 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243.

[27] K. Lu, X.R. Meng, W.X. Sun, et al. GRU-based encoder-decoder for short-term CHP heat load forecast, IOP Conf. Ser.: Mater. Sci. Eng. 392 (2018), 062173. https://doi.org/10.1088/1757-899x/392/6/062173.

[28] K. Cho, B. van Merrienboer, C. Gulcehre, et al. Learning phrase representations using RNN encoder–decoder

for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014: pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179.

[29] H. Yousuf, M. Lahzi, S.A. Salloum, et al. A systematic review on sequence-to-sequence learning with neural network and its models, Int. J. Electric. Computer Eng. 11 (2021), 2315-2326. https://doi.org/10.11591/ijece.v11i3.pp2315-2326.

[30] S. Du, T. Li, Y. Yang, et al. Multivariate time series forecasting via attention-based encoder–decoder framework, Neurocomputing. 388 (2020), 269–279. https://doi.org/10.1016/j.neucom.2019.12.118.

[31] N. Bjorck, C.P. Gomes, B. Selman, et al. Understanding batch normalization. In: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, (eds.) Advances in Neural Information Processing Systems 31, pp. 7694–7705. Curran Associates, Inc. (2018).

[32] L.N. Smith, A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay, (2018). http://arxiv.org/abs/1803.09820.

[33] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015. https://doi.org/10.48550/arXiv.1412.6980.

[34] I. Jebli, F.Z. Belouadha, M.I. Kabbaj, et al. Prediction of solar energy guided by pearson correlation using machine learning, Energy. 224 (2021), 120109. https://doi.org/10.1016/j.energy.2021.120109.

[35] Y. Liu, Y. Mu, K. Chen, et al. Daily activity feature selection in smart homes based on pearson correlation coefficient, Neural Process. Lett. 51 (2020), 1771–1787. https://doi.org/10.1007/s11063-019-10185-8.

[36] D. Chicco, M.J. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, PeerJ Computer Sci. 7 (2021), e623. https://doi.org/10.7717/peerj-cs.623.

[37] M.V. Shcherbakov, A. Brebels, N.L. Shcherbakova, et al. A survey of forecast error measures, World Appl. Sci. J. 24 (2013), 171–176.

[38] P. Schober, C. Boer, L.A. Schwarte, Correlation coefficients, Anesthesia Analgesia. 126 (2018), 1763–1768. https://doi.org/10.1213/ane.0000000000002864.

[39] J.F. Hair, C.M. Ringle, M. Sarstedt, PLS-SEM: Indeed a silver bullet, J. Market. Theory Practice. 19 (2011),

139–152. https://doi.org/10.2753/mtp1069-6679190202.

[40] T.W. Cenggoro, F. Tanzil, A.H. Aslamiah, E.K. Karuppiah, B. Pardamean, Crowdsourcing annotation system of object counting dataset for deep learning algorithm, IOP Conf. Ser.: Earth Environ. Sci. 195 (2018), 012063. https://doi.org/10.1088/1755-1315/195/1/012063.

[41] B. Pardamean, H.H. Muljo, T.W. Cenggoro, et al. Using transfer learning for smart building management system, J. Big Data. 6 (2019), 110. https://doi.org/10.1186/s40537-019-0272-6.

[42] A. Chauhan, Time series data mining for solar active region classification, 2017. https://doi.org/10.13140/RG.2.2.15327.05283.

[43] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger, KQ (eds) Advances in neural information processing systems. Curran Associates, Inc, (2014), 3104–3112.

[44] S. Park, M.S. Gil, H. Im, et al. Measurement noise recommendation for efficient kalman filtering over a large amount of sensor data, Sensors. 19 (2019), 1168. https://doi.org/10.3390/s19051168.