

Article

A Machine Learning Model Based on GRU and LSTM to Predict the Environmental Parameters in a Layer House, Taking CO₂ Concentration as an Example

Xiaoyang Chen ^{1,2,3}, Lijia Yang ¹, Hao Xue ^{1,2,3} , Lihua Li ^{1,2,3} and Yao Yu ^{1,2,3,4,*}¹ College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding 071001, China² Key Laboratory of Meat and Layer Breeding Facilities Engineering, Hebei Agricultural University, Baoding 071001, China³ Hebei Provincial Key Laboratory of Livestock and Poultry Breeding Intelligent Equipment and New Energy Utilization, Hebei Agricultural University, Baoding 071001, China⁴ Research Centre of Information Technology, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China

* Correspondence: yuyaoa2009@gmail.com

Abstract: In a layer house, the CO₂ (carbon dioxide) concentration above the upper limit can cause the oxygen concentration to be below the lower limit suitable for poultry. This leads to chronic CO₂ poisoning in layers, which manifests as listlessness, reduced appetite, weak constitution, decreased production performance, and weakened resistance to disease. Regulating ventilation may ensure a suitable CO₂ concentration in layer houses. Predicting the changes in CO₂ concentration and regulating the CO₂ concentration in advance are key to ensuring healthy large-scale breeding of layers. In recent years, machine learning and deep learning methods have been increasingly applied to this field. A CO₂ prediction model for layer house is proposed based on a GRU (gated recurrent unit) and LSTM (long short-term memory). The temperature, humidity, and CO₂ were determined as inputs to the model by the correlation coefficient. The datasets of the experimental layer house were continuously measured during June–July 2023, using a self-developed environmental monitor, and the monitored data were used as samples for model inputs. There were 22,000 time series data in the datasets. In this study, multivariate time series data were standardized via data pre-processing to improve model training. GRU and LSTM models were constructed. The models were trained using a training set. Then, these trained models were used to provide predictions on a test set. The prediction errors were calculated using the true values of the test set and the predicted values provided by the models. To test the performance of the model and accuracy of the predictions, predictions were made for different numbers of datasets. The results demonstrated that the combined prediction model had good generalization, stability, and convergence with high prediction accuracy. Due to the structure of the model, the stability of the LSTM model was higher than that of the GRU model, and its prediction accuracy and speed were lower than those of the GRU model. When the datasets of the GRU model were 15,000 to 17,000, the MAE of the GRU was 70.8077 to 126.7029 ppm, and the prediction time of the GRU is 16 to 24 ms. When the LSTM model's datasets were 15,000–20,000, the MAE of LSTM was 78.8596 to 136.0896 ppm, and the prediction time of the GRU was 17 to 26 ms.



Citation: Chen, X.; Yang, L.; Xue, H.; Li, L.; Yu, Y. A Machine Learning Model Based on GRU and LSTM to Predict the Environmental Parameters in a Layer House, Taking CO₂ Concentration as an Example. *Sensors* **2024**, *24*, 244. <https://doi.org/10.3390/s24010244>

Academic Editors: Adnan Shahid, Eli De Poorter and Jaron Fontaine

Received: 26 October 2023

Revised: 23 December 2023

Accepted: 25 December 2023

Published: 31 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: layer house; CO₂ concentration; prediction; GRU; LSTM

1. Introduction

A closed layer house with stacked cages is a relatively closed environment. Thus, the environmental factors in the house, such as temperature, relative humidity, and harmful gases and dust concentrations, can substantially impact the growth, development, reproduction, production, and health of the poultry [1]. The harmful gases produced in a layer house not only pollute the environment but also affect the physical health of layers or

suppress their production potential, resulting in economic loss in the farming industry. These harmful gases mainly include NH_3 (ammonia), H_2S (hydrogen sulfide), and CO_2 , which can be produced at harmful concentrations. Some of these harmful gases do not have a distinct odor and do not pose any serious hazards for short exposures and when at low concentrations. However, some gases damage health at high concentrations and result in acute symptoms, but this is less likely to occur under poultry production conditions. Long-term exposure to low concentrations of harmful gases causes progressive harm to both human staff and poultry health and reduces poultry productivity, causing the health status of the poultry to be poor, resulting in reduced immunity and reduced production performance. The symptoms of this condition can be easily confused with those of malnutrition or invasion by pathogenic microorganisms. Consequently, determining whether the poultry condition is due to the effects of long-term exposure to low concentrations of harmful gases can be difficult. Therefore, in modern farming and production, the air quality problem in livestock housing must be addressed [2]. In [3], it is reported that CO_2 emission linearly increases with increasing layer age in days during broiler farming and production. Higher concentrations of NH_3 and CO_2 in animal barns can negatively affect production and health of animals and workers. Additionally, indoor and ambient environmental conditions such as temperature and relative humidity were measured simultaneously with pollutant gas concentrations [4].

Pollutants emitted from poultry housing facilities are a concern from human health, bird welfare, and environmental perspectives. Development of emission factors for these aerial pollutants is difficult due to variable climatic conditions, the number and type of poultry, and the wide range of management practices used [5].

In recent years, machine learning and deep learning methods have been increasingly applied to this field.

At present, prediction models for time series include SVM (support vector machine), ARIMA (autoregressive integrated moving average), and RF (random forest). The computational complexity of the SVM algorithm is relatively high, especially for large-scale and high-dimensional datasets, which require a large amount of computation time and space. In addition, the training process of the SVM algorithm requires multiple iterations, which also increases the computational complexity. The RF algorithm is more complex, computationally expensive, and requires more time to train than other algorithms [6]. The ARIMA model requires complete time series data with no missing values. In practical applications, there are often missing and incomplete data, which affects the effectiveness of the model [7,8].

The prediction method based on recurrent neural networks has advantages in terms of prediction accuracy and speed. In [9], a novel PIHNN (physics-informed hybrid neural network) model based on the LSTM neural network is proposed. In [10], Chinese stock returns were modeled and predicted using LSTM. The historical data of the Chinese stock market were transformed into 30-day-long sequences with 10 learning features and 3-day earning rate labeling. The model was fitted by training on 900,000 sequences and tested using another 311,361 sequences. In [11], a CNN (convolutional neural network)–LSTM model is proposed to predict gas field production based on a gas field in southwest China. The CNN has a feature extraction ability, and LSTM can learn sequence dependence. By the combination of the two abilities, the CNN–LSTM model can describe the changing trend of gas field production. In [12], augmentation of fully convolutional networks with LSTM–RNN (recurrent neural) submodules for time series classification is proposed. The proposed LSTM–FCN (fully convolutional network) achieves state-of-the-art performance compared with others. In [13], a novel traffic forecast model is proposed based on an LSTM network. Differently from conventional forecast models, the proposed LSTM network considers temporal–spatial correlation in a traffic system via a two-dimensional network that is composed of many memory units. A comparison with other representative forecast models validated that the proposed LSTM network can achieve better performance. In [14],

a GRU network is proposed based on experience replay and the Snake Optimizer for real-time prediction in real-world non-stationary channels.

At present, there are relatively few studies on the prediction of carbon dioxide concentration in small layer houses in China. In addition, most existing studies rely on single-model prediction.

In this study, LSTM and GRU models with good prediction performance were constructed to predict the carbon dioxide concentration in a small layer house.

The second part of the paper introduces the acquisition of the datasets, the research of GRU and LSTM networks, and the construction of the prediction models. The third part draws the curve between the predicted and actual values of the model, draws the MAE and MSLE curves of the model prediction, and analyzes the performance of the GRU and LSTM models. The fourth part discusses the research of the paper. The fifth part is the conclusion of the paper.

2. Materials and Methods

2.1. Data Acquisition

The experiment was conducted in a layer house from January to December in 2023. The selected experimental layer house was an intensive closed layer house. Each layer house was 100 m long, 15 m wide, and 4 m high, with five rows and four layers of cage equipment. The breed of the layers was Hailan grey, with a feeding density of 74 layers per group, and the designed feeding capacity of a single building was 53,280 layers. The layer house utilized a combined positive and negative pressure ventilation system with a longitudinal design. The positive pressure ventilation system consisted of 95 fans with a diameter of 300 cm each, and the negative pressure ventilation system consisted of 18 fans with a diameter of 1400 cm each. The house was equipped with wet curtains, covering a total area of 119 m². The monitoring device consisted of a data acquisition node, master control module, and remote terminal management platform and continuously collected data from 00:00 to 24:00 every day. There were 12 devices in the layer house, from A to M. The laying hen house and the monitoring device in the layer house are shown in Figures 1 and 2.

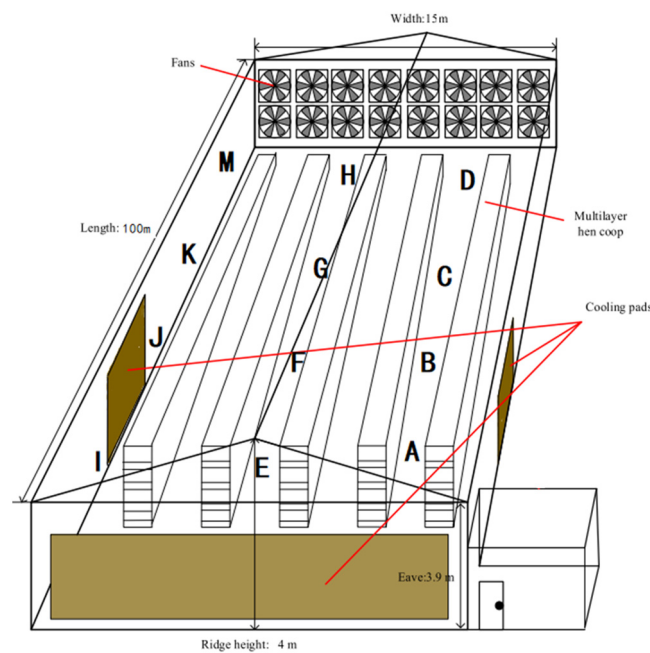


Figure 1. The on-site image of the measurement experiment in the layer house.

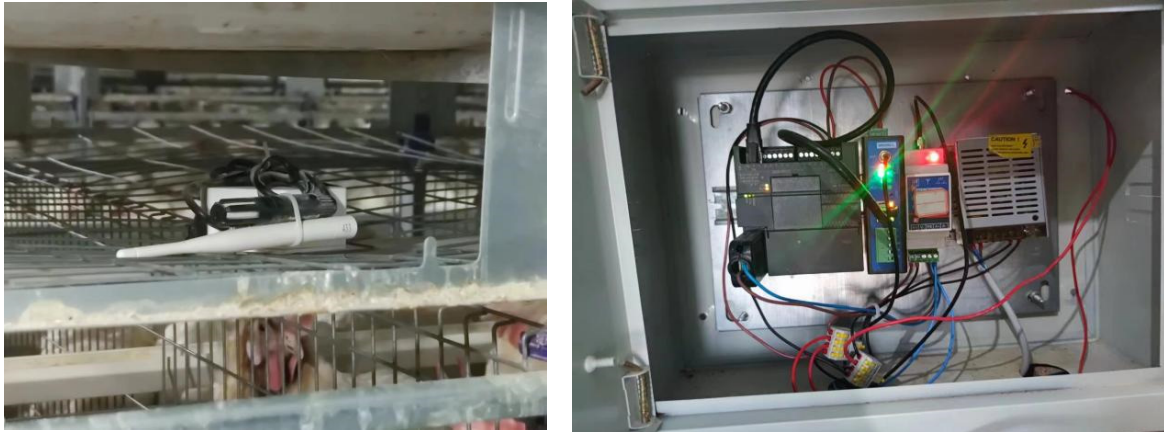


Figure 2. The temperature monitoring device in the laying hen house. The measurements lasted from May to June 2023.

2.2. Correlation Coefficient

The layer house CO₂ concentration exhibits time-series patterns and autocorrelation, that is, the data from the previous moment influences the current change of data [15]. Therefore, a model should not only consider the influence of real-time environmental factors but also the influence of historical data on the current moment [16].

To improve the effectiveness of the model input information and reduce irrelevant noise interference, this study adopted the correlation coefficient to analyze the degree of influence of the factors on the layer house CO₂ concentration and selected the factors with high correlation as the model's inputs [17]. The correlation coefficients were calculated using Equation (1).

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (1)$$

The correlation coefficient is a method for evaluating the correlation between two variables based on a monotonic function with a range of $(-1, 1)$. The closer its absolute value is to 1, the stronger the correlation. The relationship between its value and degree of correlation is shown in Figure 3.

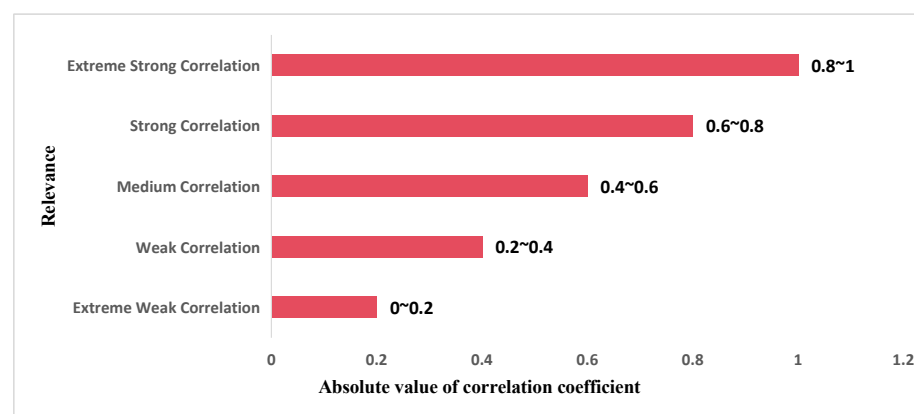


Figure 3. Relationship between absolute value of correlation coefficient and degree of correlation.

In this study, the correlation coefficient was used to estimate the correlation between the temperature, humidity, and CO₂ concentration. The results are shown in Figure 4.

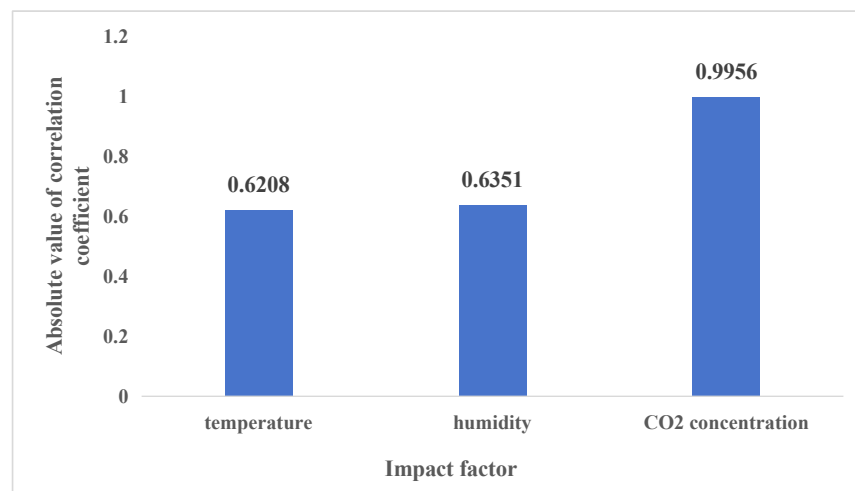


Figure 4. The correlation coefficient between the predicted value of carbon dioxide concentration and the historical values of the environmental parameters.

As shown in Figure 4, historical temperature and humidity were strongly correlated with the real-time CO₂ concentration, and historical CO₂ concentration was extremely strongly correlated with the real-time CO₂ concentration.

This study only considered factors that were at least strongly correlated with the CO₂ concentration. Therefore, temperature, humidity, and CO₂ concentration were selected as inputs for the model.

2.3. LSTM

LSTM is an RNN [18,19] with special structures that do not suffer from vanishing and exploding gradient problems during training with long sequences. In summary, LSTM performs better on longer sequences than regular RNNs do and is not affected by the gradient attenuation problems that occur with RNNs [13]. A graphical representation of the structure of LSTM is shown in Figure 5.

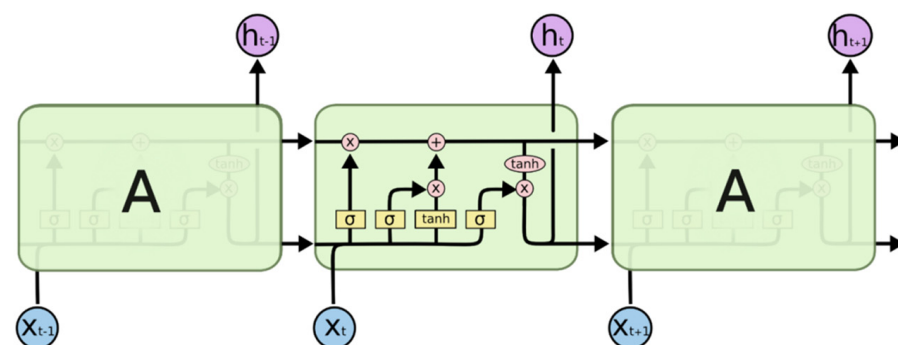


Figure 5. Graphical representation of the structure of LSTM.

Memory, referred to as a cell that has the same shape as in the hidden state, is introduced into LSTM to record additional information. Input, output, and forget gates are included to control the memory [20].

2.3.1. Forget Gate

A forget gate decides the types of information that should be discarded or retained. The information from both the previous hidden state and the current input is simultaneously transmitted to a sigmoid function. The output value is between zero and one. The information should be discarded or retained when the output value is closer to zero or one, respectively [21]. The forget gate is shown in Equation (2).

$$f_t = \sigma(x_t w_{xf} + h_{t-1} w_{hf} + b_f) \quad (2)$$

2.3.2. Input Gate

An input gate is used to update the state of the cell. First, the information of the previous hidden state, h_{t-1} , and the information of the current input, X_t , are transmitted to a sigmoid function [22]. The value is adjusted between zero and one to determine the types of information to be updated, with zero representing unimportant and one representing important information [23]. Second, h_{t-1} and X_t are transmitted to a tanh function to create a vector with a new candidate value (candidate memory). Finally, the output value of the sigmoid function, I_t , is multiplied by the output value of the tanh function, which is the candidate memory [24]. The output value of the sigmoid function determines the types of information in the output value of the tanh function that are important and need to be retained [25]. The input gate is shown in Equation (3).

$$i_t = \sigma(x_t w_{xi} + h_{t-1} w_{hi} + b_i) \quad (3)$$

2.3.3. Output Gate

An output gate is used to determine the value of the next hidden state. A hidden state contains information from previous inputs [26]. First, h_{t-1} and X_t are transmitted to a sigmoid function; second, the new cell state, C_t , is transmitted to the tanh function [27]. Finally, the output of the tanh function is multiplied by the output of the sigmoid function, O_t , to determine the types of information that the hidden state should carry, h_t [28]. The hidden state is then used as the output of the current cell, transmitting C_t and the new hidden state, h_{t+1} , to the next time step [29]. The output gate is shown in Equation (4).

$$o_t = \sigma(x_t w_{xo} + h_{t-1} w_{ho} + b_o) \quad (4)$$

2.4. GRU

Gated recurrent neural networks are designed to better capture the dependencies of time series data with larger intervals. The GRU is a commonly used gated recurrent neural network [26].

In the hidden layers of RNNs, the gradients of variables may vanish or explode. Although gradient clipping can cope with the exploding gradient problem, it cannot resolve the vanishing gradient problem. The GRU, similar to LSTM, was developed to overcome problems such as the gradient in long-term memory and backpropagation [30].

The input and output structures of a GRU are the same as those of a regular RNN. The input includes both the input at moment t , x_t , and the hidden layer state at moment $t - 1$, h_{t-1} , which contain the information about the previous node. The output includes the output of the hidden node at moment t , y_t , and the hidden state to be transmitted to the next node, h_t [31].

Unlike LSTM, which has three gates, GRU has only two gates, which are the reset and update gates. The states of the two gates are obtained from the previous transmitted state, h_{t-1} , and the input of the current node, X_t [32]. A graphical representation of the structure of a GRU is shown in Figure 6.

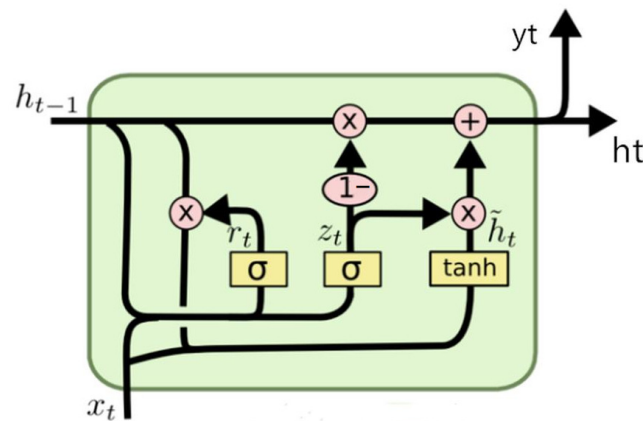


Figure 6. Graphical representation of the structure of a GRU.

2.4.1. Update Gate

An update gate, r_t , is used to control the extent to which the state information from the previous moment is applied in the current state, with more state information being used from the previous moment as the update gate value increases. The information from the previous and current moment is right-multiplied by a weight matrix. Subsequently, the summed data are fed into the update gate and multiplied by a sigmoid function to yield a value between zero and one [33]. The update gate is shown in Equations (5) and (6).

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t]) \quad (5)$$

$$\tilde{h}_t = \tanh(w \cdot [r_t * h_{t-1}, x_t]) \quad (6)$$

2.4.2. Reset Gate

A reset gate, z_t , is used to control how much historical information from the previous state is written to the current candidate set, with less information from the previous state written as the reset gate value decreases [34]. The data processing of the reset gate is the same as that of the update gate, where the information from the previous and current moments is right-multiplied by a weight matrix. Subsequently, the summed data are fed into the reset gate and multiplied by a sigmoid function to yield a value between zero and one. The differences between the two weight matrices are their values and purposes [19]. The reset gate is shown in Equations (7) and (8).

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t]) \quad (7)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (8)$$

2.5. Model Construction

(1) Hardware and software of the workstation

A workstation was set up for data processing, and the hardware configuration was as follows: CPU: AMD 7773X (AMD, Santa Clara, CA, USA), main board: GIGABYTE MZ72-HB0 (GIGABYTE, New Taipei City, Taiwan), graphics card: AMD RADEON PRO W7800 32 GB GDDR6 RDNA3 (AMD, Santa Clara, USA), ROM: 32 G DDR4 3200 RECC (KINGSTON, Fountain Valley, CA, USA), hard disk: SAMSUNG 1 TB M2 NVME (SAMSUNG, Seoul, Republic of Korea).

The software configuration was as follows, OS: Windows 10 Professional Edition (Microsoft, Redmond, WA, USA), Environmental configuration: Anaconda 3 and PyCharm Community Edition (Anaconda, Austin, TX, USA).

(2) Obtaining the datasets

The datasets for this study were obtained from an environmental monitoring system installed in the experimental layer house. The datasets took the average value of the measured data. There were 22,000 time series data in the datasets. The time series of the datasets is shown in Figure 7.

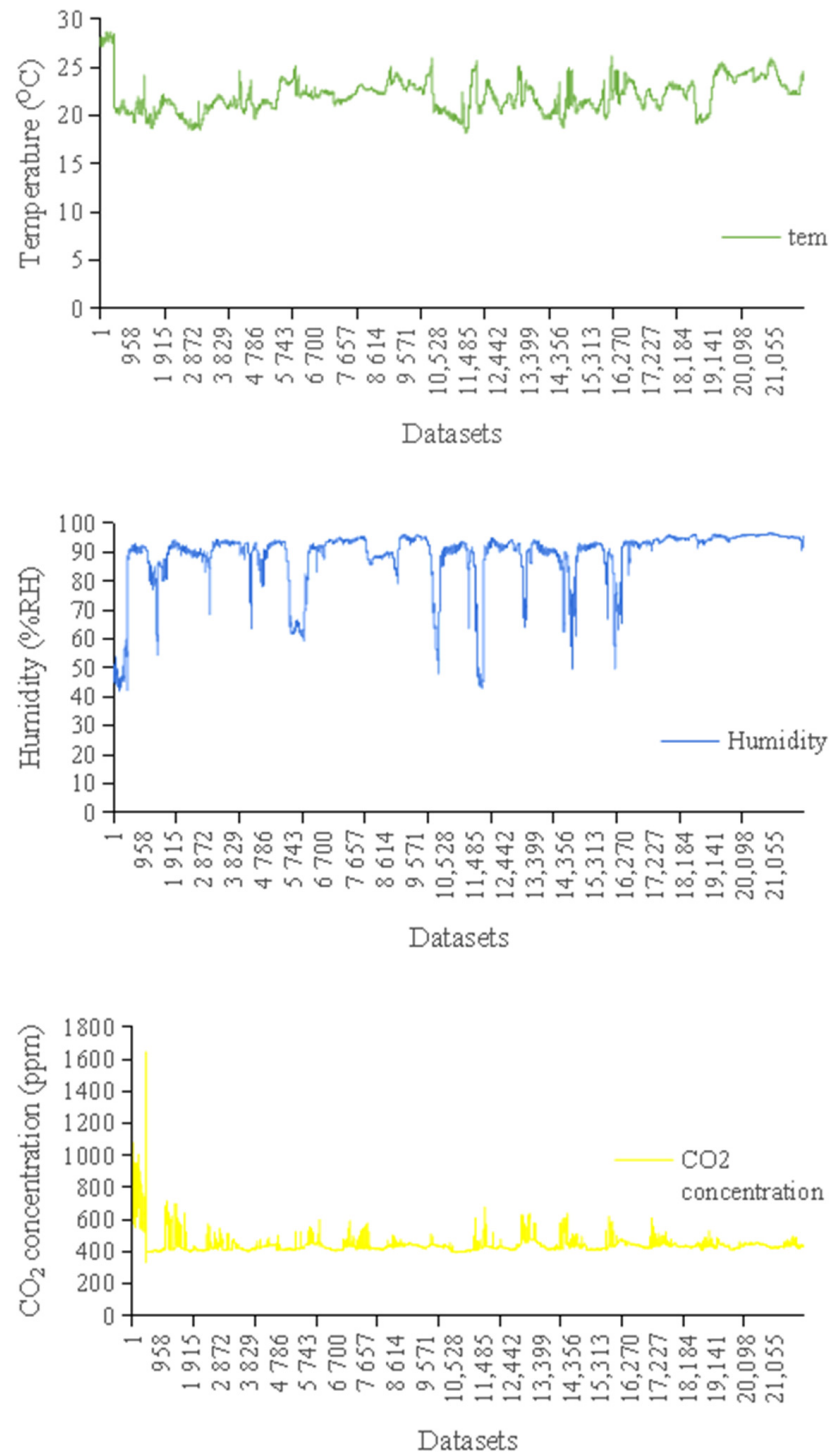


Figure 7. Time series diagrams of temperature, humidity, and CO₂ concentration in the layer house.

(3) Data pre-processing

The sklearn standardization method was used to standardize all the feature data, as standardization can help prevent data with excessive bias from affecting the training results. The standardized datasets are shown in Figure 8.

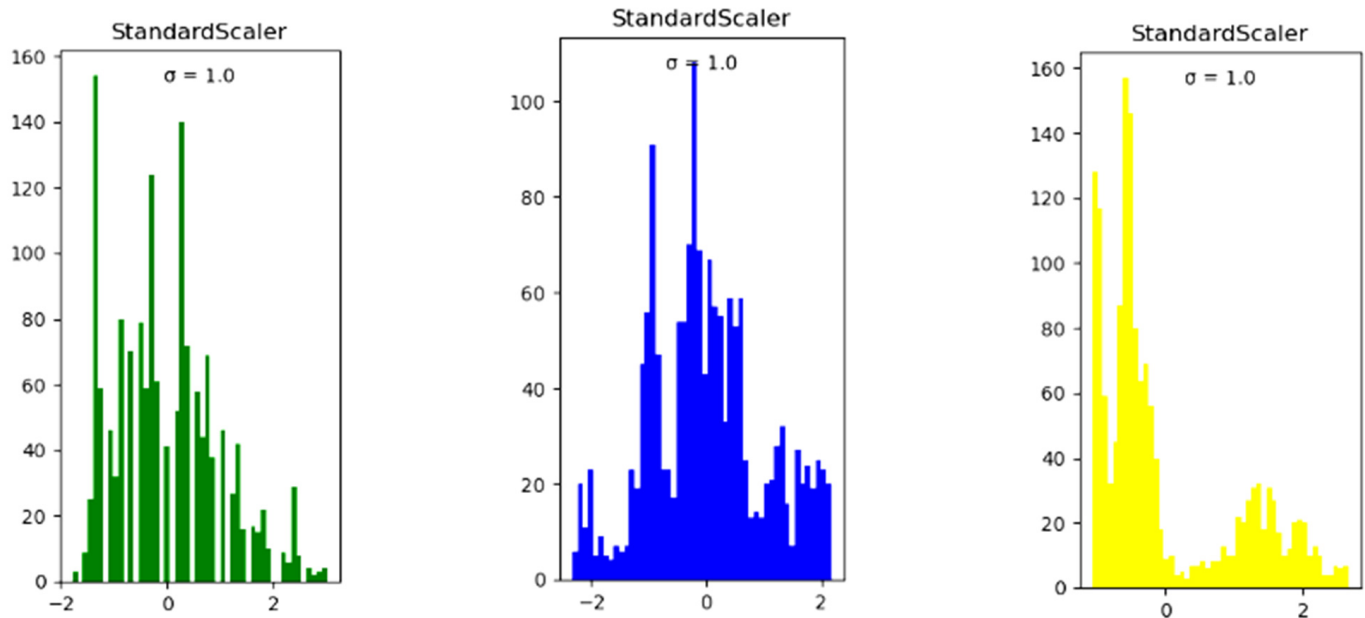


Figure 8. The standardized datasets of temperature, humidity, and CO₂ concentration.

(4) Time series: sliding window method

A convenient queue, deque, was used to specify the maximum queue length, maxlen, as 20, denoting that the time series length was 20. If queue length, deq, was more than 20, the first feature was deleted, whereas the 21st feature was appended to the 20th feature to constantly maintain a queue length of 20. As such, each time series had a shape of [5,13], denoting 20 rows of data and 5 columns of features.

After classifying the time series of all the data, the last 10 sets of the time series were deleted, as the last 10 rows of the feature data, sca_x, had no corresponding label value. Each series had a corresponding label, and the lengths of each series and its corresponding label were the same.

(5) Dataset partitioning

We obtained the processed time series and their corresponding labels. Subsequently, the training, validation, and test sets were proportionally divided. Overall, 80% of the datasets was used for training, 10% of the data was used for validation, and 10% of the data was used for testing. For the training set, shuffle(·) was used to randomly disrupt the arrangement of the data rows to avoid contingency. The iterator, iter(·), was set in conjunction with the next(·) function to retrieve a batch of data from the training set.

(6) Construction of network models

We used a GRU network as an example here. The structure of the GRU model is shown in Table 1.

Table 1. The structure of the GRU model.

Model: "GRU"		
Layer (type)	Output Shape	Parameter
input_4 (InputLayer)	[(None, 20, 5)]	0
gru_3 (GRU)	(None, 20, 8)	360
dropout_12 (Dropout)	(None, 20, 8)	0
gru_4 (GRU)	(None, 20, 16)	1248
dropout_13 (Dropout)	(None, 20, 16)	0
gru_5 (GRU)	(None, 32)	4800
dropout_14 (Dropout)	(None, 32)	0
dense_6 (Dense)	(None, 16)	528
dropout_15 (Dropout)	(None, 16)	0
dense_7 (Dense)	(None, 1)	17
Total parameters: 6953		
Trainable parameters: 6953		
Non-trainable parameters: 0		

(7) Hyperparameters:

The configuration settings of the hyperparameters for the GRU and LSTM models is shown in Table 2.

Table 2. Hyperparameters for the GRU and LSTM models.

Hyperparameters	Value
timestep	1
batch_size	32
feature_size	1
hidden_size	256
output_size	1
num_layers	2
epochs	100
best_loss	0
learning_rate	0.0003

(8) Hyperparameter optimization

This study applied the adaptive learning rate algorithm as Adam.

(9) Network training

The mean absolute error (MAE) between the predicted and label values was used as the loss function, and the mean squared logarithmic error (MSLE) was used as a metric for monitoring the networks. The MAE and MSLE were retained for each iteration during training. Early Stopping, a simple and effective method for avoiding overfitting, was used during model training. It works by monitoring the validation set performance during model training and stopping the training when the validation set performance is optimal, which can determine appropriate training rounds and avoid overfitting effectively.

(10) Inspecting training process information

For each iteration, the loss and monitoring metrics for both the training and validation sets were plotted.

(11) Forecasting

The evaluate(·) function was used to calculate the loss and monitoring metrics for the entire test set to obtain a timescale for each true value. The GRU and LSTM models' prediction flow is shown in Figure 9.

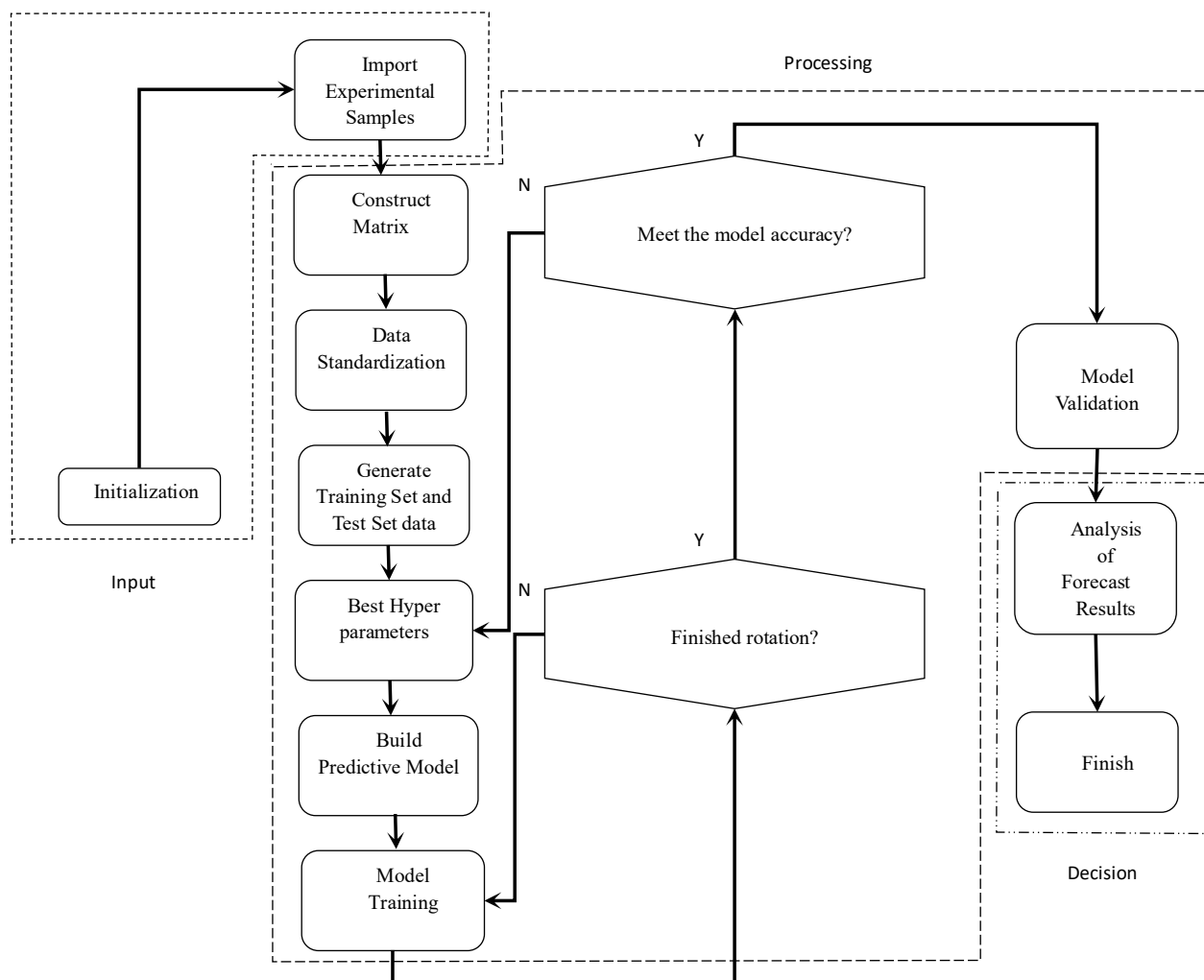


Figure 9. The experimental procedure of this study to develop the GRU and LSTM models.

3. Results

3.1. Evaluation Indicators

To evaluate the performance and assess the reliability of the models after training was completed, the input factors of the validation set were fed into the trained model to get the predicted values. These predictions were then compared to the actual observed values. Mean absolute error (MAE) between the predicted and label values was used as the loss function, and mean squared logarithmic error (MSLE) was used as a metric for monitoring the networks.

MSLE is the indicator used to evaluate the prediction model. It is the average value of logarithmic error, which can be used to measure the prediction ability of the model for data with large values. MSLE is calculated as shown in Equation (9).

$$\text{MSLE} = \frac{1}{N} \sum [\log(y_i + 1) - \log(\hat{y}_i + 1)]^2 \quad (9)$$

The range of MAE is $(0, +\infty)$, with 0 indicating a perfect model where the predicted value matches the true value exactly. The larger the error, the larger the MAE value. MAE is shown in Equation (10).

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (10)$$

3.2. Predictive Evaluation

The sample sizes of the test sets were 5000–20,000, and the predicted and actual values are shown in Figure 10.

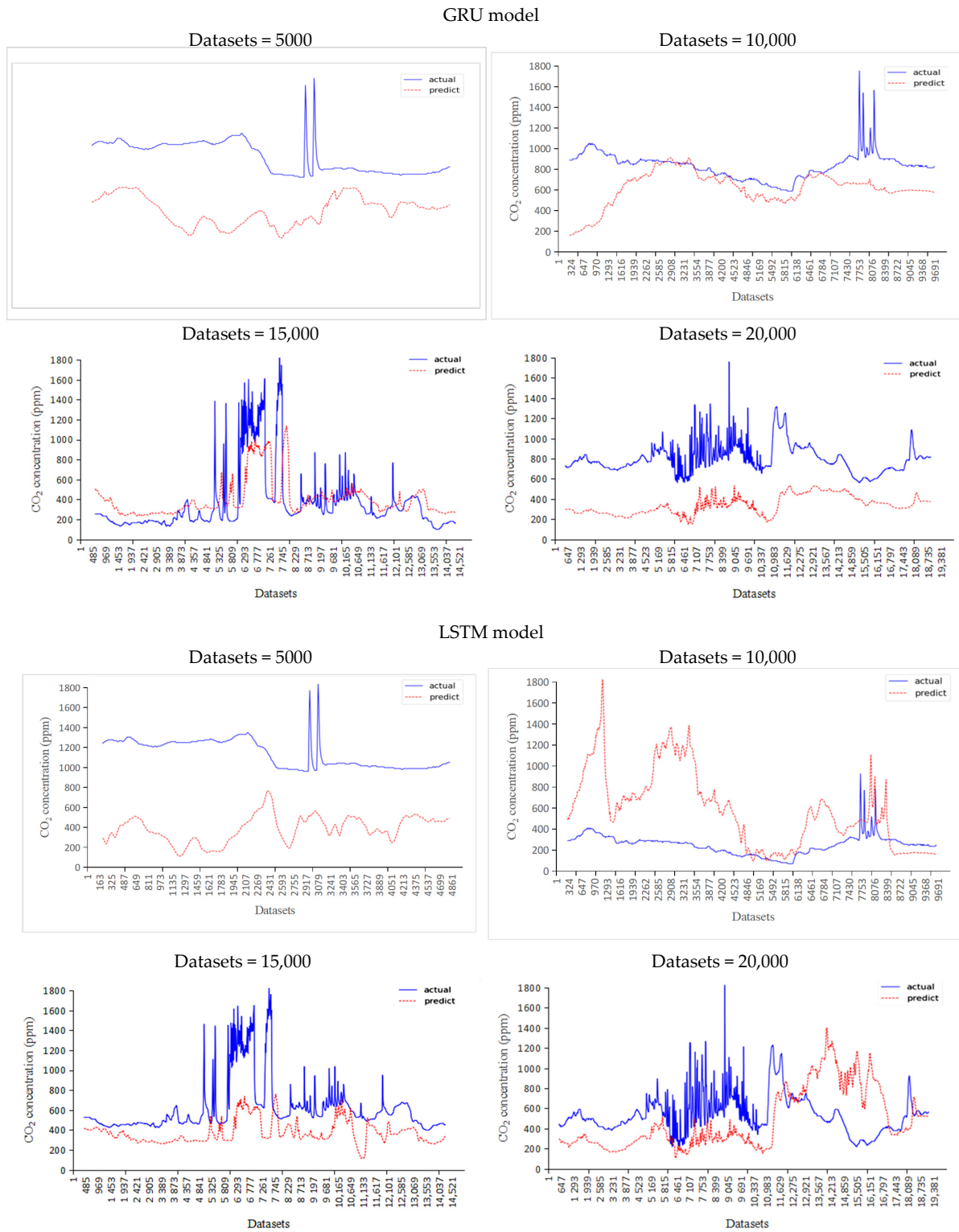


Figure 10. Comparison of predicted and actual values with the GRU and LSTM models.

It can be seen from Figure 10 that when the number of datasets put into the GRU model was increased from 5000 to 15,000, the agreement between the predicted and actual values gradually increased. When 20,000 datasets were put into the GRU model, the degree

of agreement between the predicted and actual values slightly decreased. We concluded that when the number of datasets put into the GRU model was within an appropriate range, the prediction accuracy increased with the number of datasets.

When the number of datasets input to the LSTM model was increased from 5000 to 20,000, the agreement between the predicted and actual values gradually increased.

3.3. Evaluation of Model Performance by Plotting Training and Validation Loss Curves

The training and validation loss curves are two evaluation metrics commonly used in the deep learning training process to represent the errors of a model during training and validation.

3.3.1. The Train_Loss and Train_Msle of the GRU Model and LSTM Model

Training loss refers to the model's error on the training set, indicating the magnitude of the model's prediction error on the training samples during the training process.

The train_loss of the GRU and LSTM models is shown in Figure 11.

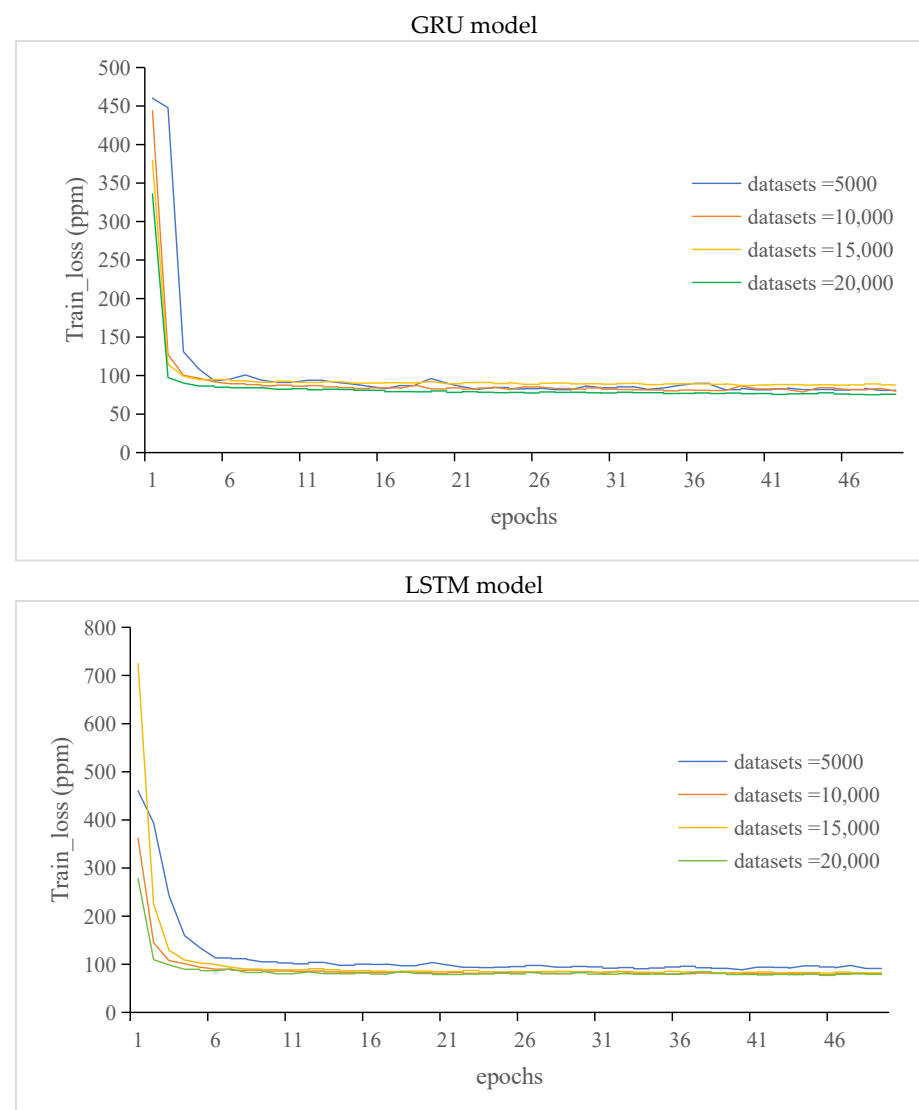


Figure 11. The train_loss of the GRU and LSTM models.

It can be seen from Figure 11 that when 5000 datasets were put into the models, the training_loss curve of the GRU model showed that the training error stayed at the maximum for one training cycle, whereas the curve of the LSTM model rapidly decreased

after training started. During model training, the validation set was used to test the performance of the models at regular intervals to obtain the val_loss results.

The training_loss of the GRU and LSTM models gradually decreased as the number of training sessions gradually increased. The models produced more accurate predictions on the training samples because they had learned more features and patterns. Therefore, the training_loss curve showed a decreasing trend as the number of training sessions increased.

The train_msle of the GRU and LSTM models is shown in Figure 12.

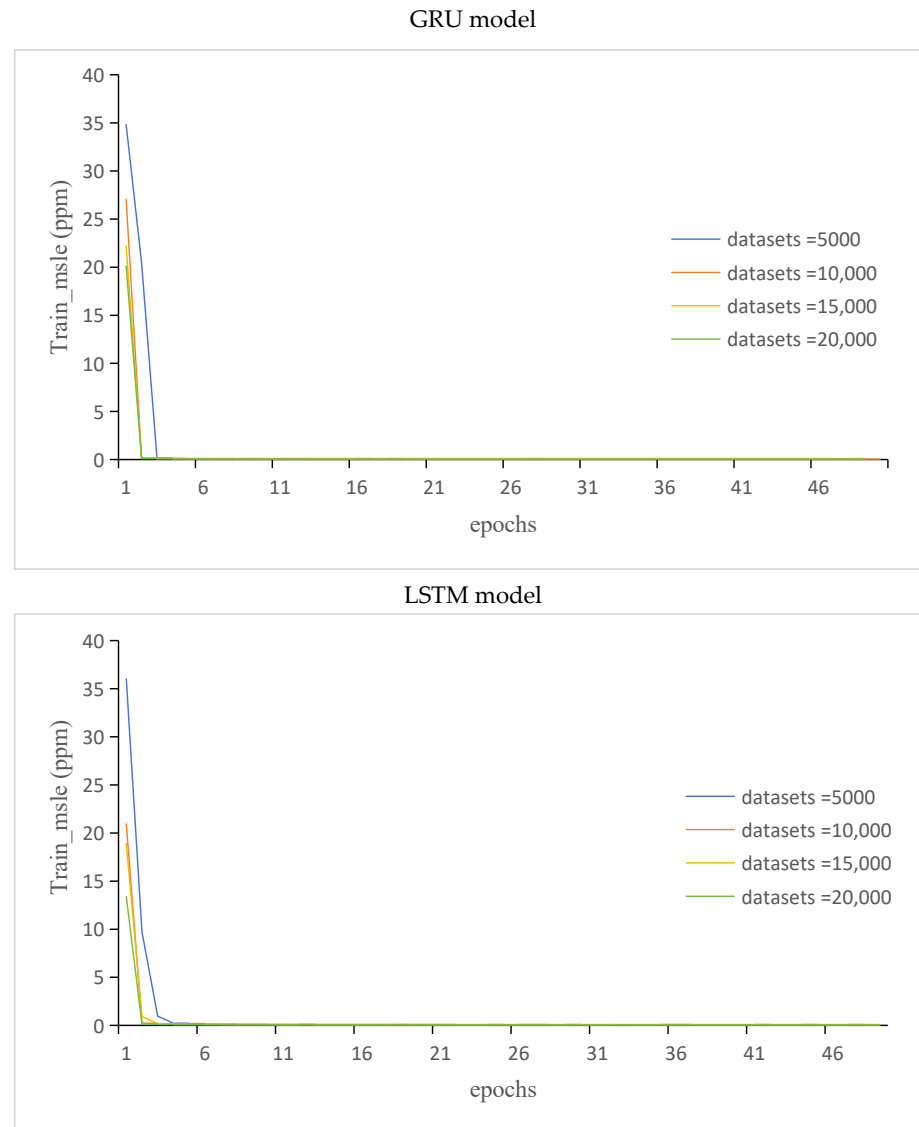


Figure 12. The train_msle of the GRU and LSTM models.

3.3.2. The Val_Loss and Val_Msle of GRU Model and LSTM Model

Validation loss refers to the model's error on the validation set, indicating the magnitude of the model's prediction error on the unknown data during the training process. The validation set accounts for a portion of the data randomly cleaned from the training set, which includes the data that the models had not yet encountered during the training process. The validation set is used to evaluate the ability of a model to predict unknown data.

The val_loss of the GRU and LSTM models is shown in Figure 13.

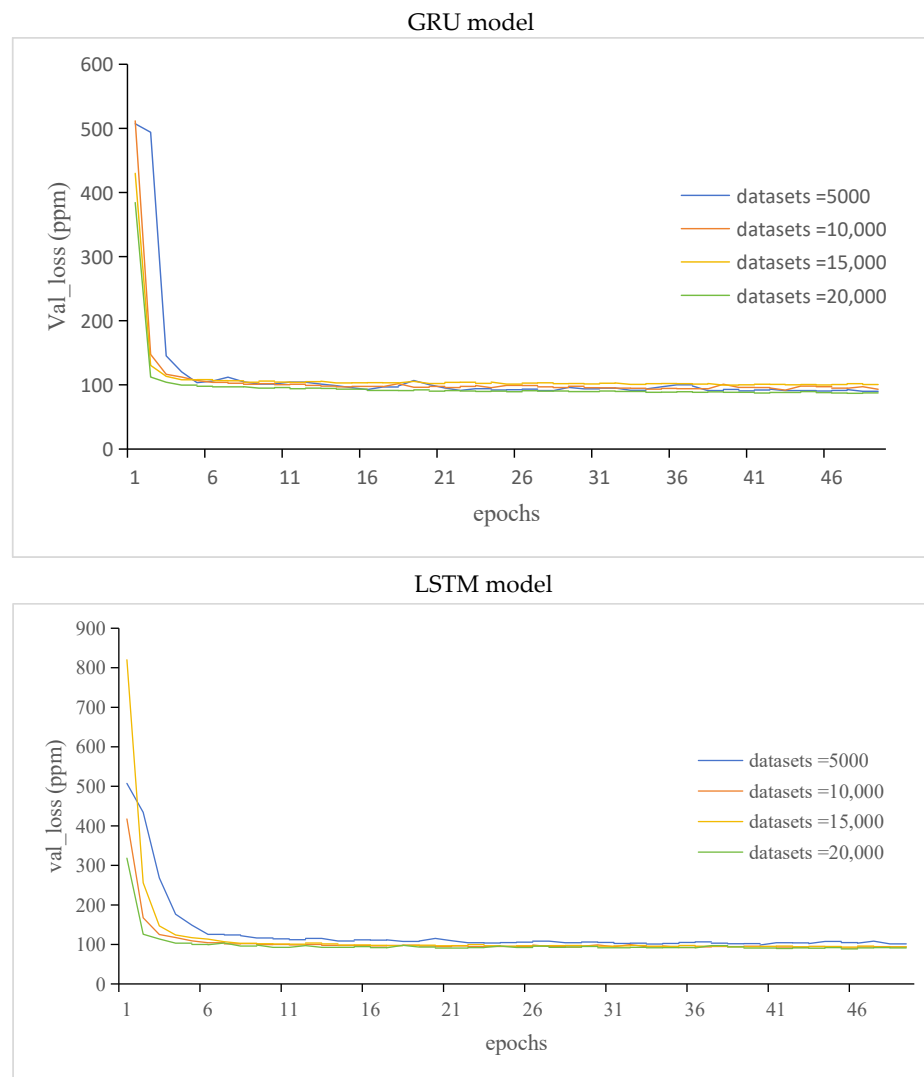


Figure 13. The val_loss of the GRU and LSTM models.

The val_msle of the GRU and LSTM models is shown in Figure 14.

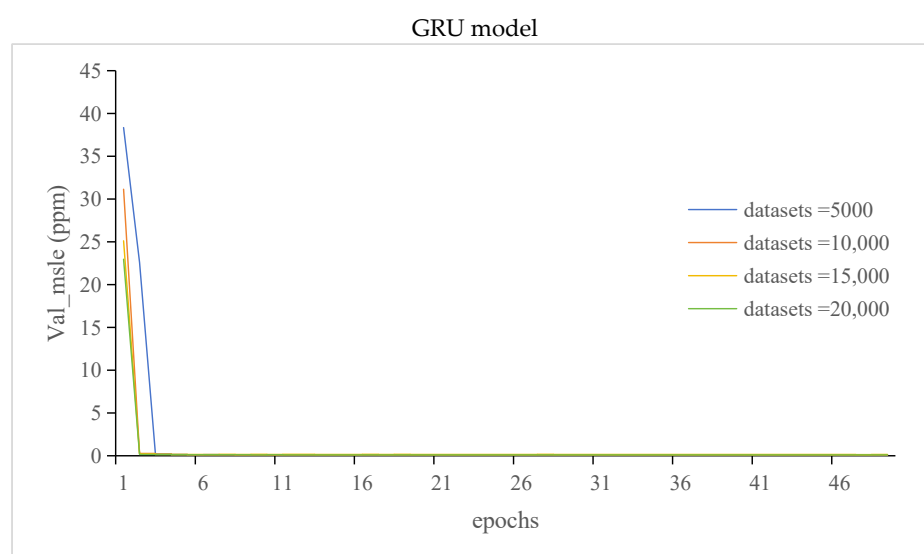


Figure 14. Cont.

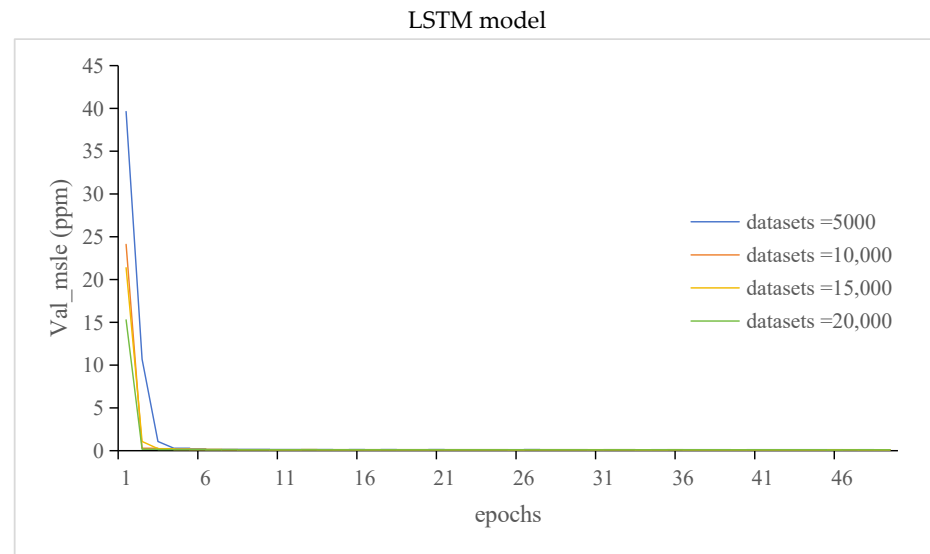


Figure 14. The val_msls of the GRU and LSTM models.

3.3.3. Overfitting Check by Cross-Validation

This study adopted Five-Fold cross-validation. We divided the datasets into five non-overlapping sub-datasets and then performed five model trainings and validations. Every time, we validated the model using one large dataset and trained the model using the other four large datasets. In these five training and validation sessions, the datasets used to validate the model were different each time. Finally, we calculated the averages of training loss and validation loss for these five iterations. The Five-Fold cross-validation is shown in Figure 15.

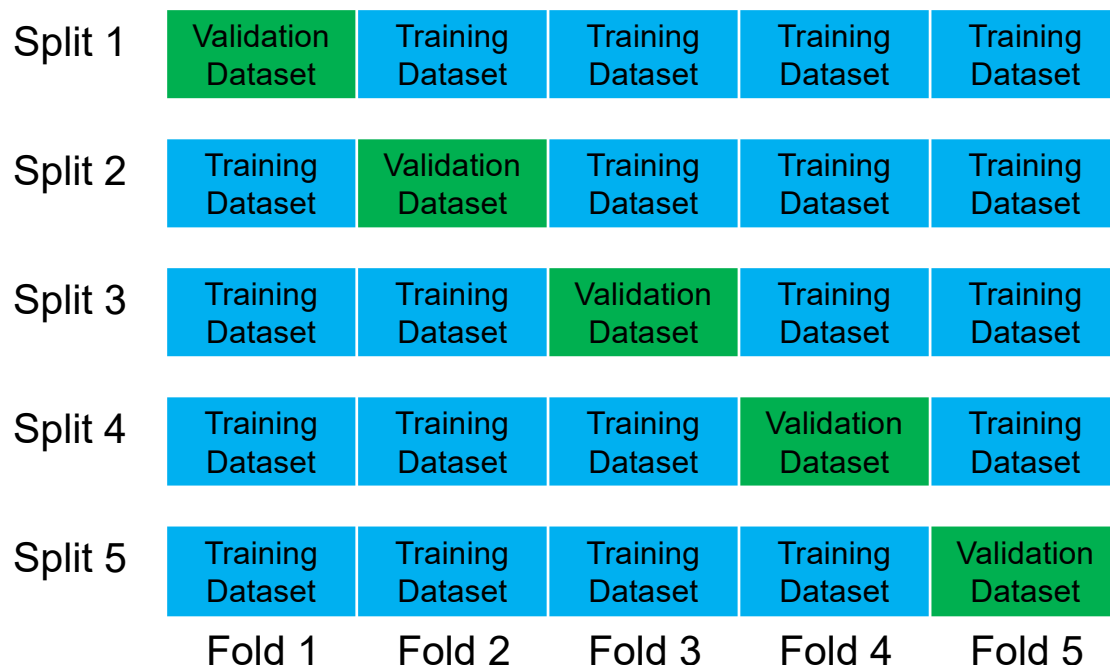


Figure 15. Five-Fold cross-validation to check for overfitting.

A comparison was made between the train_loss and val_loss of the LSTM model and the GRU model to determine if there was overfitting in the model. The comparison results are shown in Table 3.

Table 3. Comparison of training error and testing error.

Model	Datasets	Train_Loss	Val_Loss
LSTM	5000	90.8156	91.6452
	10,000	88.0604	89.8628
	15,000	86.0255	87.3469
	20,000	84.8316	85.3657
GRU	5000	80.7534	82.4733
	10,000	78.8040	80.3956
	15,000	76.4618	78.3562
	20,000	79.3265	80.6643

It can be seen from Table 3 that when the train_loss of the LSTM model and GRU decreased, val_loss also decreased, and there was no overfitting in the model.

3.3.4. Model Application

A comparative field test for the prediction model was conducted in an experimental layer house, in where the actual CO₂ concentration in the layer house was used as the model input to predict the CO₂ concentration after one minute.

The predicted and actual values of the CO₂ concentration were considered as the basis for the ventilation of the layer house, and the upper limit of CO₂ concentration was set as 500 ppm.

If the predicted value of the CO₂ concentration was used as the basis for ventilation, the fan was turned on when the predicted value reached 500 ppm.

If the actual value of the CO₂ concentration was used as the basis for ventilation, the fan was turned on when the actual value reached 500 ppm.

The CO₂ concentration and fan status of the layer house in both cases are shown in Table 4.

Table 4. CO₂ concentration and fan operation state inlayer house under different ventilation mechanism.

Fan Opening Basis	Actual Value and Measuring Time of Carbon Dioxide Concentration in Layer House	Predict Value and Corresponding Time of Carbon Dioxide Concentration in Layer House	Status of Fan
Predicted value of carbon dioxide concentration	495.8376 ppm 09:10:25	499.2168 ppm 09:11:25	Off
	495.3592 ppm 09:10:40	500.0410 ppm 09:11:40	Off
	499.9687 ppm 09:11:40	503.6828 ppm 09:12:40	On
	499.2547 ppm 09:11:55	495.3158 ppm 09:12:55	Off
Actual value of carbon dioxide concentration	499.6548 ppm 09:10:40		Off
	500.1364 ppm 09:11:40		On
	500.0129 ppm 09:11:50		On
	499.3159 ppm 09:12:00		Off

Table 4 shows that if the predicted value was used as the basis, at 9:11:40, the predicted and actual values were 500.0410 ppm and 499.9687, respectively, and the fan was turned on. At 09:11:55, the actual value was 499.2547 ppm, and the fan was turned off.

If the actual value was used as the basis, when the actual value was 500.1364 ppm, the fan was turned on. At 09:12:00, the actual CO₂ concentration was 499.3159 ppm, and the fan was turned off.

It could be judged from the experimental results that if the predicted value was used as the basis for ventilation, the CO₂ concentration in the layer house could be ensured to remain below the limit value at all times, and the total operation time of the fan was 15 s.

If the actual value was used as the basis for ventilation, the CO₂ concentration exceeded the limit value for 10 s after starting the fan, and the total operation time of the fan was 20 s.

This concludes that using the predicted value as the basis for adjusting the operating status of fans could reduce the CO₂ concentration faster and ensure the constraints of the CO₂ concentration in the chicken house.

4. Discussion

It can be seen from Figure 10 that when the number of datasets of the GRU and LSTM models gradually increased, the agreement of the predicted and actual values gradually increased.

However, when the number of datasets reached 20,000, the agreement of the predicted and actual values of the GRU model decreased, while the agreement of the LSTM model did not decrease.

It could be concluded that the model had optimal datasets. When the number of datasets was within the optimal range, the agreement of the predicted and actual values increased with number of datasets. When the number of datasets exceeded the optimal range, the agreement of the predicted and actual values was uncertain.

In order to acquire the optimal number of datasets, prediction was attempted with 15,000 to 20,000 datasets, with 1000 datasets added each time.

The MAE of the GRU and LSTM is shown in Figure 16.

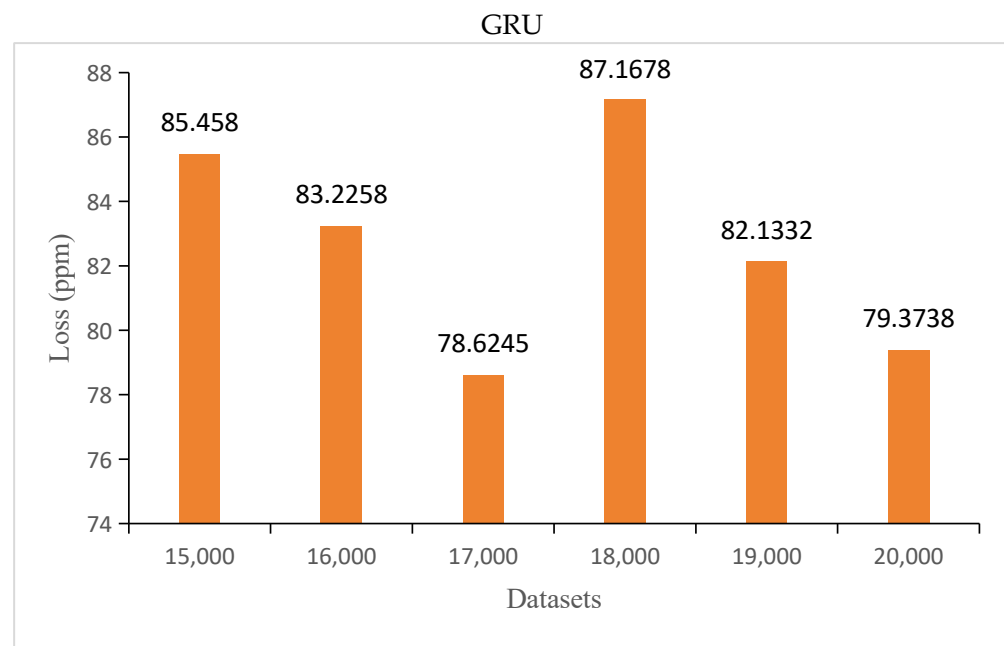


Figure 16. Cont.

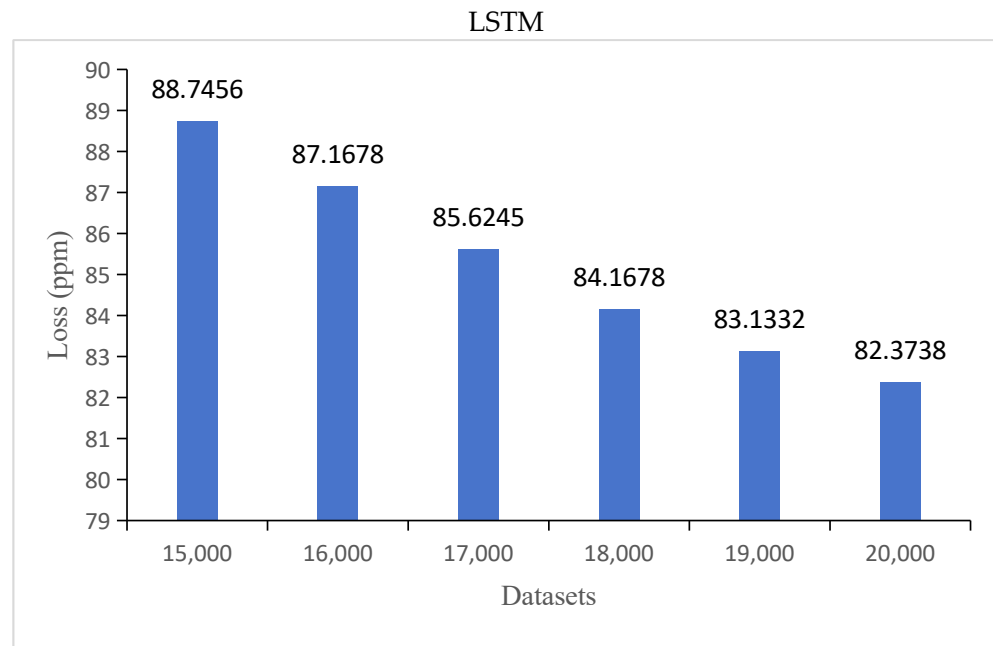


Figure 16. Loss of the GRU and LSTM models within 15,000 to 20,000 datasets.

It can be seen from Figure 16 that when the number of datasets was between 15,000 and 17,000, the prediction loss of the GRU model decreased with the increase of the number of datasets. When the number of datasets exceeded 17,000, the agreement between the predicted and actual values was uncertain. It could be concluded that the optimal number of datasets for the GRU model was between 15,000 and 17,000. When the number of datasets was between 15,000 and 20,000, the prediction loss of the LSTM model decreased with increasing number of datasets, and the agreement between the predicted and real values was certain. It could be concluded that the optimal number of datasets for the LSTM model was between 15,000 and 20,000.

It can be seen from Figures 11–14 that compared with those of the GRU model, the LSTM model has more stable but lower prediction speed and more loss. The reason was that the two models had different structures. The LSTM model contained three gates the, forget gate, the input gate, and the output gate, and the LSTM model had more parameters. The GRU model contained two gates, the update gate and the reset gate. Due to the structure of the model, the stability of the LSTM model was higher than that of the GRU model, and the prediction accuracy and speed were lower than those of the GRU model.

The loss of the GRU model was 70.8077 to 126.7029 ppm, and the loss of LSTM model was 78.8596 to 136.0896 ppm. The prediction time of the GRU model was 16 to 24 ms, and the prediction time of the LSTM model was 17 to 26 ms.

When the model was applied in a layer house, the measured value of the carbon dioxide concentration in the house was used as the input, which required high accuracy of the layer house carbon dioxide measurement device. The layer house manager needs to invest in hardware procurement costs when using this model, which is applicable to managers of large-scale farming.

5. Conclusions

- (1) In this study, an extensive examination of the performance of GRU and LSTM models in predicting environmental parameters, specifically the CO₂ concentration in a layer house, provided significant insights and contributions to the field of predictive modeling for agricultural and environmental applications.
- (2) According to the correlation coefficients, the layer house temperature, humidity, and CO₂ concentration were selected as the feature data, and a model for predicting the CO₂ concentration in a layer house was constructed based on the GRU and LSTM.

- (3) Different datasets were selected, and the corresponding prediction results were obtained. The training and validation errors of the GRU and LSTM models were analyzed. The results showed that there was an optimal range of the number of datasets for the prediction model, and the loss of the model was minimal within this range
- (4) MAE between the predicted and label values was used as the loss function, and MSLE was used as a metric for monitoring the networks. MAE and MSLE were retained for each iteration during training. For each iteration, the loss and monitoring metrics of both the training set and validation set were plotted. The evaluate(·) function was used to calculate the loss and monitoring metrics for the entire test set to obtain a timescale for each true value.
- (5) While increasing the dataset size yielded an improvement in prediction accuracy for both GRU and LSTM models, the findings noted a decline in prediction accuracy for the GRU model when 20,000 datasets were utilized. This suggests that the GRU model's performance might plateau or decline beyond a certain dataset threshold, indicating a limitation in handling larger datasets. We will focus on solving this problem in future work.
- (6) The datasets for this study were collected between June and July 2023. In future research, the collection time of the datasets will be extended to four seasons to improve the applicability and robustness of the model.
- (7) This study will address the computational efficiency issue of the model in future work.

Author Contributions: Conceptualization, X.C., Y.Y. and L.L.; methodology, X.C. and L.Y.; validation, Y.Y. and H.X.; formal analysis, X.C.; investigation, X.C.; resources, X.C.; data curation, X.C.; writing—original draft, X.C.; writing—review and editing, X.C., Y.Y. and L.L.; visualization, X.C.; supervision, X.C.; funding acquisition, Y.Y. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the S&T Program of Hebei (22326607D), Modern agricultural industrial technology system in Hebei Province Innovation Team Project (HBCT2023210204), Beijing Innovation Consortium of Digital Agriculture Research System. (BAIC10-2023). Research Project on Basic Research Business Expenses of Provincial Higher Education Institutions in Hebei Province (KY2021017).

Institutional Review Board Statement: This study used historical data from normal production records at the farm, which did not cause additional harm to laying hens and therefore did not require an ethics approval.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Q.; Liu, P.; Liu, Z.; An, L.; Wang, M.; Wu, Z. Main Factors Influencing the Ammonia Emission of Poultry House. *China Poult.* **2017**, *39*, 45–49. [[CrossRef](#)]
2. Zhao, Y. Effects of Carbon Dioxide Concentration on Performance, Blood Biochemical Indexes and Nutrient Metabolism of Laying Hens. Master's Thesis, Hebei Agricultural University, Baoding, China, 2018.
3. Burns, R.T.; Li, H.; Xin, H.; Gates, R.S.; Overhults, D.G.; Earnest, J.; Moody, L.B. *Greenhouse Gas (GHG) Emissions from Broiler Houses in the Southeastern United States*; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2008; p. 1. [[CrossRef](#)]
4. Kilic, I.; Yaslioglu, E. Ammonia and carbon dioxide concentrations in a layer house. *Asian-Australas. J. Anim. Sci.* **2014**, *27*, 1211. [[CrossRef](#)] [[PubMed](#)]
5. Morgan, R.J.; Wood, D.J.; Van Heyst, B.J. The development of seasonal emission factors from a Canadian commercial laying hen facility. *Atmos. Environ.* **2014**, *86*, 1–8. [[CrossRef](#)]
6. Taheri, S.; Razban, A. Learning-based CO₂ concentration prediction: Application to indoor air quality control using demand-controlled ventilation. *Build. Environ.* **2021**, *205*, 108164. [[CrossRef](#)]

7. Ma, C.F.; Xie, Q.J.; Wang, S.C. A prediction model for environmental factors in a pig house based on statistical learning and deep learning. *J. Shanxi Agric. Univ. Nat. Sci. Ed.* **2022**, *42*, 24–32. [[CrossRef](#)]
8. Baghban, A.; Bahadori, A.; Mohammadi, A.H.; Behbahaninia, A. Prediction of CO₂ loading capacities of aqueous solutions of absorbents using different computational schemes. *Int. J. Greenh. Gas Control.* **2017**, *57*, 143–161. [[CrossRef](#)]
9. Zarzycki, K.; Ławryńczuk, M. Long Short-Term Memory Neural Networks for Modeling Dynamical Processes and Predictive Control: A Hybrid Physics-Informed Approach. *Sensors* **2023**, *23*, 8898. [[CrossRef](#)]
10. Chen, K.; Zhou, Y.; Dai, F. A LSTM-based method for stock returns prediction: A case study of China stock market. In Proceedings of the 2015 IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2823–2824. [[CrossRef](#)]
11. Zha, W.; Liu, Y.; Wan, Y.; Luo, R.; Li, D.; Yang, S.; Xu, Y. Forecasting monthly gas field production based on the CNN-LSTM model. *Energy* **2022**, *260*, 124889. [[CrossRef](#)]
12. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM fully convolutional networks for time series classification. *IEEE Access* **2017**, *6*, 1662–1669. [[CrossRef](#)]
13. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [[CrossRef](#)]
14. Liu, Q.; Wang, P.; Sun, J.; Li, R.; Li, Y. Wireless Channel Prediction of GRU Based on Experience Replay and Snake Optimizer. *Sensors* **2023**, *23*, 6270. [[CrossRef](#)] [[PubMed](#)]
15. Asuero, A.G.; Sayago, A.; González, A.G. The correlation coefficient: An overview. *Crit. Rev. Anal. Chem.* **2006**, *36*, 41–59. [[CrossRef](#)]
16. Taylor, R. Interpretation of the correlation coefficient: A basic review. *J. Diagn. Med. Sonogr.* **1990**, *6*, 35–39. [[CrossRef](#)]
17. van Aert, R.C.; Goos, C. A critical reflection on computing the sampling variance of the partial correlation coefficient. *Res. Synth. Methods* **2023**, *14*, 520–525. [[CrossRef](#)] [[PubMed](#)]
18. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *40*, 132306. [[CrossRef](#)]
19. Shewalkar, A.; Nyavanandi, D.; Ludwig, S.A. Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 235–245. [[CrossRef](#)]
20. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv* **2015**. [[CrossRef](#)]
21. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; pp. 324–328. [[CrossRef](#)]
22. Shahid, F.; Zameer, A.; Muneeb, M. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos Solitons Fractals* **2020**, *140*, 110212. [[CrossRef](#)]
23. Zhang, Z.; Robinson, D.; Tepper, J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web, Proceedings of the 15th International Conference, ESWC, Heraklion, Greece, 3–7 June 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 745–760. [[CrossRef](#)]
24. Athiwaratkun, B.; Stokes, J.W. Malware classification with LSTM and GRU language models and a character-level CNN. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2482–2486. [[CrossRef](#)]
25. Gao, S.; Huang, Y.; Zhang, S.; Han, J.; Wang, G.; Zhang, M.; Lin, Q. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. *J. Hydrol.* **2020**, *589*, 125188. [[CrossRef](#)]
26. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A tutorial into long short-term memory recurrent neural networks. *arXiv* **2019**. [[CrossRef](#)]
27. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 799–804. [[CrossRef](#)]
28. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [[CrossRef](#)]
29. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [[CrossRef](#)]
30. Yang, S.; Yu, X.; Zhou, Y. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In Proceedings of the International Workshop on Electronic Communication and Artificial Intelligence (IWEC AI), Shanghai, China, 12–14 June 2020; pp. 98–101. [[CrossRef](#)]
31. Agarap, A.F.M. A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In Proceedings of the 10th International Conference on Machine Learning and Computing, Macau, China, 26–28 February 2018; pp. 26–30. [[CrossRef](#)]
32. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600. [[CrossRef](#)]

33. Bansal, T.; Belanger, D.; McCallum, A. Ask the gru: Multi-task learning for deep text recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 107–114. [[CrossRef](#)]
34. Sajjad, M.; Khan, Z.A.; Ullah, A.; Hussain, T.; Ullah, W.; Lee, M.Y.; Baik, S.W. A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. *IEEE Access* **2020**, *8*, 143759–143768. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.