AgriEngineering

MDPI

Article

Machine-Learning Microclimate Forecasting for Adaptive Equipment Control via Web Integration in Open-Ventilated Greenhouses

Kyaw Maung Maung Thwin [1], Teerayut Horanont [1,*] and Teera Phatrapornnant [2]

[1] School of ICT, Sirindhorn International Institute of Technology, Thammasat University, Bangkadi Muang, Pathum Thani 12000, Thailand; m6522040648@g.siit.tu.ac.th
[2] National Electronic and Computer Technology Center, Khlong Luang District, Pathum Thani 10210, Thailand; teera.phatrapornnant@nectec.or.th
* Correspondence: teerayut@siit.tu.ac.th

Abstract: Open-ventilated greenhouses have reasonable setup costs and low operational costs for growers, which is crucial and most appealing for this research. These attributes fit developing nations like Thailand and other tropical regions. It is challenging to control the equipment intended to obtain an ideal microclimate. This research was conducted in an actual greenhouse setting for data collection and experiments, with a proposed system for adaptive equipment control via web integration. Also, the proposed multivariate multistep LSTM was forecasted over 1 h and cooperated with sensor data. Additional sensors, like a leaf wetness sensor and a CO2 sensor, were installed for detecting plant-level precision for vaporization, rather than greenhouse-level. The proposed system can optimize the indoor temperature within 34.5 to 36 °C with a 39 to 40 °C outdoor temperature. Also, humidity was still at the ideal level of 68 to 70%; more precisely, the wetness value was below 300 throughout the experiment. The model accuracy achieved a sufficient RMSE (0.49) and R^2 (0.9788). This proposed system architecture and mm-LSTM model has potential as one dimension of a fully smart greenhouse system development in open-ventilated greenhouse settings in tropical regions and Southeast Asian nations for a better yield rate and less human interaction.

Keywords: open-ventilated greenhouse; multivariate multistep LSTM; web integration; adaptive equipment control

check for updates

Citation: Thwin, K.M.M.; Horanont, T.; Phatrapornnant, T. Machine-Learning Microclimate Forecasting for Adaptive Equipment Control via Web Integration in Open-Ventilated Greenhouses. *AgriEngineering* **2024**, *6*, 2845–2869. https://doi.org/10.3390/agriengineering6030165

Academic Editor: Simone Pascuzzi

Received: 19 July 2024
Revised: 8 August 2024
Accepted: 9 August 2024
Published: 13 August 2024

Copyright: © 2024 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/).

1. Introduction

Greenhouses can be categorized into three main types for this research, based on their level of enclosure and ventilation and how the greenhouse microclimate interacts with the outside environment, which can directly impact the microclimate and suitability of different crops [1]. Here are the details of the three categories of greenhouses. The first is a fully enclosed greenhouse, which is enclosed entirely from the external environment with glass, polycarbonate, or other rigid materials, for roofs and walls, to construct the structure and embedded with environmental control systems such as air conditioning, heating, ventilation, artificial lighting, and drip irrigation systems to maintain excellent growing conditions throughout the plantation. This greenhouse has many advantages, such as complete control over the microclimate environmental parameters, being more resilient to pests and diseases, year-round production capability, protection from external contamination, and maximizing the yield and quality of the plants by controlling the ideal microclimate [2]. High initial investment and operational costs would be the most significant burden for farmers and growers to set up a fully enclosed greenhouse for practical usage [3].

Secondly, a semi-enclosed greenhouse combines the ventilation approach of a fully enclosed greenhouse and an open-ventilated greenhouse. Microclimate control systems

integrate the mechanism of adjusting the vents and windows opening and closing to regulate temperature and humidity, reducing the reliance on energy-intensive heating, ventilation, and air conditioning (HVAC) systems, which is one of the essential benefits of semi-enclosed greenhouses [4]. Technically, this type of greenhouse can act as a fully or open-ventilated greenhouse in terms of ventilation. However, it can be easily exposed to pest control and disease risks compared to a fully enclosed greenhouse. This greenhouse's disadvantages are high system complexity and maintenance of the actuator and equipment due to the adjustable roof, windows, and equipment [5].

Thirdly, open-ventilated greenhouses are notable for their utilization of natural ventilation to create natural air circulation; as required, mechanical equipment is equipped in this greenhouse to optimize the microclimate and prevent outdoor climate dependency. The main advantage of this open-ventilated greenhouse is a lower initial setup cost compared to other types of greenhouses and energy efficiency, as well as minimizing the need for mechanical cooling and ventilation by leveraging natural ventilation, which can reduce operational costs significantly [6]. In research conducted in the literature, countries like Thailand and other Southeast Asian nations and tropical regions can benefit from setting up this type of greenhouse, since farmers and growers cannot afford high investments without international funding and projects. On the other hand, the most challenging part is climate dependency in an open-ventilated greenhouse, since natural airflow has an effect on the microclimate of the greenhouse, so maintaining optimal growing conditions is up to controlling the equipment. However, behaviors of dramatic changes in patterns of outdoor climate require monitoring and intervention for equipment control.

In a South Korean fully enclosed greenhouse, temperature, humidity, and CO_2 were predicted by using three different neural network architectures, an artificial neutral network, nonlinear autoregressive exogenous model, and RNN-LSTM (recurrent neutral network–long short-term memory), and research prioritized the accuracy of the prediction performance and thus evaluated using 5 to 30 min prediction lengths [7]. Likewise, an unmentioned greenhouse-type environmental parameter prediction was researched using GCP-LSTM and a 5 min data processing window sliding approach, and different types of crops were experimented with [8]. In Mexico, a ventilated greenhouse experiment with RNN-LSTM, with varying arrangements for feeding, modeled a one-hour prediction and experimented with five different arrangements of three parameters in an even-ventilated greenhouse. Still, none of the equipment was installed indoors [9]. Defining the setpoint is one of the challenging things for greenhouse environmental parameters; the LightGbm algorithm was used to determine the setpoint with the integration of Neural Language Processing (NLP) and an expert hybrid system in a simulated greenhouse, which mimicked the collated data from an actual greenhouse [10]. Handling multiple parameters of greenhouse environmental reduction by using the Principal Component Analysis Method (PCA) from 15 to 7 parameters, which still represented 99.19%, one researcher proposed improved support vector regression using the parameter reducing method to make a prediction that integrated with the PSO algorithm for optimization, where the system was used in an experiment in tomatoes cultivated in a fully enclosed greenhouse [11].

Open-ventilated greenhouse-type temperature and humidity optimization research was conducted in Indonesia, close to our environmental conditions. It experimented with water spray on and off conditions without considering any machine-learning or prediction techniques, with nozzle rotation to 60 degrees showing greater effectiveness [12].

The operational cost of a greenhouse with DDRMPC (data-driven robust model predictive control) with the integration of dynamic control models is reduced by 4% compared to conventional rule-based control [13]. Dealing with a decent humidity level was important; it was proportional to temperature reduction in the greenhouse indoors because higher vaporization can cause fungus and plant diseases, so the greenhouse installed the leaf wetness sensor for more precise detection of vaporization [14]. A greenhouse farm in the Netherlands conducted experiments in a fully enclosed greenhouse with five teams for five smart greenhouses. The research pointed out the importance of the plants receiving

sufficient illumination and trimming, as well as the fact that not all additional sensors effectively improve yield rates [15]. However, the previous research on precision greenhouse systems mentioned above did not sufficiently consider open-ventilated greenhouses. Moreover, some research was conducted on weather conditions that differed from those in Southeast Asia.

More details on the open-ventilated type of greenhouse regarding machine-learning applications need to be provided. This research focused on open-ventilated greenhouse microclimate control and was conducted in Thailand, which is located in a tropical region and is a developing nation like other Southeast Asian countries.

Open-ventilated greenhouses are quickly affected by outdoor weather conditions and cannot fully control environmental parameters. This can lead to a decrease in yield rate and quality unless growth is consistently monitored. Mechanical equipment like fans, water spray, and shading were installed inside the greenhouse. Controlling this equipment was one of the challenging factors in achieving the plant's ideal microclimate level. That led to significant human interaction requirements, and time spent for them could detract from other aspects of the business or essential tasks that might be more beneficial. High humidity and high-temperature countries like Thailand face the drawback of having unfavorable climate conditions for indoor plantations. The main objective of this research is to address the microclimate in the open-ventilated greenhouses to obtain the ideal temperature and humidity by using a data-driven precision greenhouse system that handles time-series data analysis with machine learning. Two main questions were identified as requirements: What would be an effective way to control the actuators and equipment like fans, mists, foggers, and others in the open-ventilated greenhouse to maintain microclimate parameters like temperature and humidity? Which data-driven approach is best suited to control the equipment during critical hours (daytime) on environmental parameters by integrating with machine-learning techniques and sensor data? Possibly answering these questions and developing the system led to this research.

This paper is arranged as follows: Section 2 explains the details of the experimental setting in an open-ventilated greenhouse and sensors. Additionally, the architecture of the multistep multivariate–long short-term memory (MM-LSTM) model and the data preprocessing steps and model training are mentioned, and then, in detail, the architecture of the greenhouse equipment control and model deployment system is discussed. Section 3 describes the entire experiment result in detail to show the mm-LSTM and the capability of the developed system. Moreover, the evaluation results presented on the customized mm-LSTM model are compared to the baseline mm-LSTM model with the test dataset and unseen experimental days dataset. Section 4 discusses the experiment results as well as the effectiveness of the proposed mm-LSTM model with the equipment control system and a general comparison with previous findings, and Section 5 concludes on the research result and tangibility of the proposed system, giving a brief explanation of this research method and the requirements of innovative open-ventilated greenhouses in precision agriculture as future work.

2. Materials and Methods

2.1. Experiment Setup and Data Acquisition

The experiment was set up in an open-ventilated greenhouse at the Faculty of Science and Technology, Thammasat University, Rangsit campus, located at 14°04′26.2″ N 100°36′31.8″ E. It had six greenhouses, and one greenhouse (Figure 1) was used for this research data collection and the experimental period. This research only focused on soil-based cultivation settings in an open-ventilated greenhouse setting. However, the plantation was not considered due to the time limitation of the research.



Figure 1. Experimental greenhouse structure from outside.

The greenhouse equipment included three fans on the top right side, two indoor fans facing each other, a 0.5 mm top spray, 0.5 mm side spray, and 0.3 mm top spray. The 0.5 mm and 0.3 mm sizes refer to the nozzle size of the spray. Indoor temperature, humidity, and illumination sensors (Figure 2a) were installed in that greenhouse, and the sensor data were collected in a database. However, this sensor installation was performed by another organization, and additional sensors were installed for our newly developed system, which is intended to achieve more precision.



(**a**) (**b**)

Figure 2. Temperature, humidity, and illumination sensor (**a**) and CO_2 sensor (**b**).

Additional sensors include carbon dioxide (CO_2) and leaf wetness sensors. CO_2 sensors (Figure 2b) are an essential tool for measuring CO_2 concentrations for many crops, which can significantly enhance photosynthesis and encourage faster and healthier plant growth. Obtaining adequate CO_2 can result in higher crop yields and better crop quality.

Knowing the CO_2 level indoors has the benefit of balancing the microclimate to reduce the risk of poor air quality and humidity levels, which are associated with causing plant diseases. Moreover, consistent CO_2 can contribute to uniform plant growth, promoting quality assurance of crops and commercial cultivation [16].

Leaf wetness sensors (Figure 3) were additionally installed to detect the presence and duration of vaporization on the leaf's surface; these sensors have many advantages, such as the early detection of leaf surface moisture if the grower uses an irrigation method of spraying over the plant. These sensors can provide real-time data and prevent unnecessary water usage, fungi, and diseases caused by vaporization. Managing the moisture levels and preventing fungi and diseases can contribute to healthier plants and higher yield rates, especially since some plants are not as favorable to water as others. Therefore, it is better to prevent over-vaporization conditions so crops can achieve better market prices and reduce post-harvest losses [17]. Since measuring solely relative humidity was insufficient for an open-ventilated greenhouse, precision moisture level detection at the plant leaf level was required.



(**a**) (**b**)

Figure 3. Leaf wetness sensor: (**a**) side view (**b**) front view.

The central control unit controls all equipment used in this experiment. It is equipped in the greenhouse and mainly controls the equipment in two ways: via a semi-automated web portal and manually at the side of the greenhouse.

Data collection was a cornerstone of the machine-learning model's performance and accuracy and hugely impacted our model performance. The data collection period started on 22 February 2024 and ended on 1 May 2024, a total of 70 days. Parameters included the indoor temperature, indoor humidity, indoor illumination, indoor carbon dioxide concentration, and fan and spray operation status, representing on and off operation throughout the data collection period. A total of 47,270 data points were collected, representing 2112 h in total. However, some hours were missed due to system downtime and internet connection interruptions during the data collection period; a total of 48 h were missed, and after substruction of the missed hours, the total remaining hours were 2112. The average data interval was calculated by Formula (1). The average data interval was approximately 2.7 min.

$$\Delta t_{avg} = \frac{T_m}{DT} \quad (1)$$

where,

Δt_{avg} = average data interval;

T_m = total duration of data points in minutes;

DT = total number of data points.

The main reason the leaf wetness sensor data were not included in the model training phase was related to two factors: the sensor was installed later than the data collection starting date, and the main objective of the leaf wetness sensor was to detect the vaporization so the data would be used in the experimental period.

2.2. Design of Multivariate Multistep LSTM Architecture

Long short-term memory (LSTM) is an updated recurrent neural network RNN version, which can learn long-term dependencies due to the new architecture introduced [18]. RNN had issues learning long-term temporal or sequential data dependencies due to gradient vanishing and exploding problems [19]. This was the case when selecting LSTM, as it mitigates these issues by using a unique architecture with cell states and gating mechanisms to control the flow of information. LSTM outperformed traditional RNNs in time-series forecasting and over long sequences, demonstrating the superior performance of LSTM [20] in applications like precipitation nowcasting that highlighted its capability of handling long temporal dependencies over long sequences. It had different variations of LSTM-like architecture designed for some specific purpose, and a refined version can be seen in gated recurrent units (GRU), which were introduced [21] as a variation of LSTM. However, the standard LSTM architecture was used in this research because that version has gained popularity for many problems.

The designed mm-LSTM model (Figure 4) can handle multivariate feature input and output multistep results. The three-dimensional array is represented as [samples, time steps, features]. It represents the multivariate time-series data where X1 to X6 are features described by indoor temperature, indoor humidity, indoor illumination, indoor CO_2, fan status, and spray status. Tx+i is the number of time steps. Each column represents a different feature, and the row represents different time steps. The LSTM layer uses the standard LSTM model to capture the temporal dependencies in the data. All the gates and states are part of the LSTM layer, and the output of this layer is a sequence of hidden states. The first Rectify Linear Unit (ReLU) layer applies the activation function to deal with the non-linearity of the data. This is fully connected to a dense layer that maps the LSTM output to a higher dimensional space. This helps in learning complex relationships in the data, followed by the dropout layer; this layer prevents overfitting by randomly setting a fraction of the input unit to zeros during training. This technique is one of the regularization techniques.

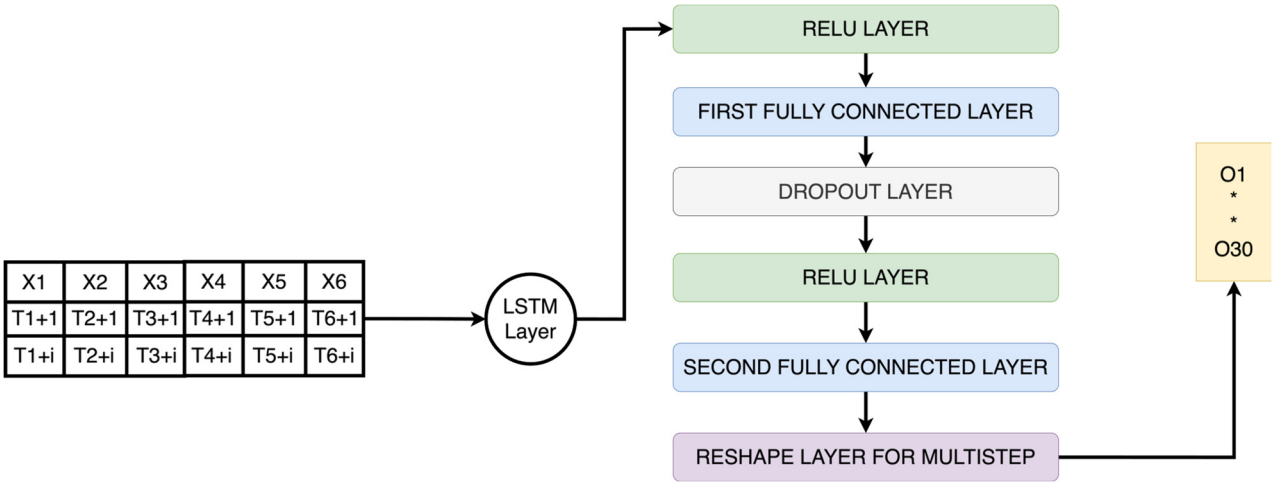


Figure 4. Multivariate multistep LSTM(MM-LSTM) architecture.

The second ReLU activation layer maintains non-linearity in the model, and the second fully connected layer is used for further data processing, which allows the model to learn additional abstract features or patterns. In the final stage, the output of the second fully

connected layer fit with the desired output shape, which was required to handle the 2D array to transform as a 1D array for each sample to match the multistep output format. (O1 ** O30) are denoted for the next 30 time steps of forecasting value over approximately 1 h. Given past observations, each output value represented a sequence of predictions for each future time step. This design for the mm-LSTM architecture could capture temporal dependencies effectively.

The LSTM model included two states and four hidden gates. The two states represent the cell state and hidden state; the four hidden gates are called the forget gate, Formula (2); input gate, Formula (4); cell gate, Formula (5); and output gate, Formula (8). The cell state also denotes long-term memory, which does not have any weights or biases so can smoothly flow until the end. The gate was handled to modify the long-term memory, including hour-difference gates. The forget gate is responsible for deciding how much information should be forgotten from the previous input, which is no longer useful. The values represent 0 and 1; 1 will take all information to the next step, and 0 otherwise. The sigmoid activation function Formula (3) rates between 0 and 1 for the forget gate.

$$f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{2}$$

$$\sigma = \frac{1}{1 + e^{-x}} \tag{3}$$

where,

σ = sigmoid function, W_f = weight coefficient of the forget gate;

h_{t-1} = hidden state, X_t = new input, b_f = bias of the forget gate.

The input gate consists of two segments, the sigmoid layer and the tanh layer, Formula (6). The sigmoid layer is responsible for updating value decisions, the vector conversion of new values, and keeping the value in this stage.

$$i_t = \sigma(W_i.[h_{t-1}, X_t] + b_i) \tag{4}$$

$$C_t = \tanh(W_c.[h_{t-1}, X_t] + b_c) \tag{5}$$

$$\text{Tanh} = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{6}$$

After two calculation stages, Formula (7) is used to additionally calculate the aggregation for the input gate.

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{7}$$

The output gate decides which value should be output as the final value, filters out the value from the cell state to the final value, uses the tanh function again, and applies Formulas (8) and (9).

$$O_t = \sigma(W_o.[h_{t-1}] + b_o) \tag{8}$$

$$h_t = O_t * \tanh(C_t) \tag{9}$$

The research model designed for multivariate multistep time-series forecasting involves multiple features as input. Each input series provides a different dimension of information, which can help to improve the model's performance accuracy by incorporating multiple input features. The model can better understand the complex relationships of the data, providing multiple future points more value for planning and decision-making [22].

2.3. Data Sequence Approach

Two types of LSTM are in the input architecture: univariate, where the input feeds one column and forecasts the output in a single step or multistep. The second type, multivariate, is used in this research, which includes multiple dataset columns [23] in tabular format, and the output can be either a single step or multiple steps. In the multivariate technique used in our research with a window-looking approach, the input values can be arranged in a way that is [Batch size, features] or [Batch size, sequence, feature], the second technique applied in our LSTM model for multistep output. This is denoted as the sliding window approach [24].

This sliding window technique is applied in the training phase, creating sub-sequences (windows) from extensive time-series data to train the LSTM model. A fixed number of time steps are contained in each window, which is used to predict the next time steps. We create input–output pairs by overlapping the data sequence formatted as a 3D array format [Batch size, sequence, features]; for example, we feed the model a sequence of data [1–7] and predict the outcome [8,9], and the next step [2–8] will target the value of [9,10] and so forth. In this way, the dataset is trained and the sequence size must remain flexible and customized as required.

Time-series data can be defined as tabular datasets and grid-like datasets. A tabular dataset is a table format where data are organized into rows and columns. Temporal data are inputted as a unique record per row, whereas columns represent data features. This data type is mainly used in matrix-type data, relational databases and spreadsheets, and homogeneous data. In contrast, grid-like data are particularly used in image processing and geographical data, which are formatted as grid-like structures and organized as cells. Each arranged cell is represented by corresponding values like surface temperature and pixel and spatial location. Examples can be found in images, digital evaluation models, and satellite data [25]. These greenhouse environmental data were formatted as tabular data because the LSTM model is more aligned with this data format than grid-like data. A multivariate multistep LSTM model feeds the 30 sequences to the model simultaneously and generates 30 sequences of forecast value.

2.4. Data Preprocessing and Model Training

The data preprocessing step is shown in Figure 5. First, the null value of the dataset is checked, and if there is any null value, the row will be removed. Secondly, the outlier of the data is checked by filtering out the outlier by eliminating temperature rates of less than 20 °C and greater than 100 °C. This is applied after analyzing the dataset behavior. Third, all duplicated rows are removed at the minute level since the sensor data are inconsistent, including overlapping minutes in data points, with granularity considered at the minute level in our model. Fourth, equipment status data are merged into the dataset using the logged fan and mist/foggy operation status. Then, finally, the dataset is saved in CSV format for the model training stage.

The model training process flow (Figure 6) included multiple steps; first, the preprocessed data were inspected to see if any null values and outliers of all features still exist; if no such thing existed, then we moved to the next step; otherwise, we iterated to the preprocessing step. In the next step, data were split into three segments: training, validation, and testing. The data were allocated as 43,321 for the training set, the validation set was 1898, and the test set was 2049. The data were split into three parts because the validation set was intended for hyperparameter tuning, so the test set would be unseen data and kept for final testing for model evaluation. As temporal data, the dataset was neither shuffled nor split in a way that allowed for constructive days. Then, the date time value was set as an index of the requirement of data frame format before standardizing the data. Then, the data were reshaped to be 1D, intended for standardization with a standard scaler Formula (10). Data were transformed with a standard scaler used by the Scikit learn library.

$$\text{SS Standaraized} = \frac{x - \mu}{\sigma} \tag{10}$$

where,

SS Standardized = standard scaler standardized value;
X = original value;
μ = mean of the features;
σ = standard deviation of the features.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{11}$$

where,

N = number of observations;
x_i = individual observation of the feature.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2} \tag{12}$$

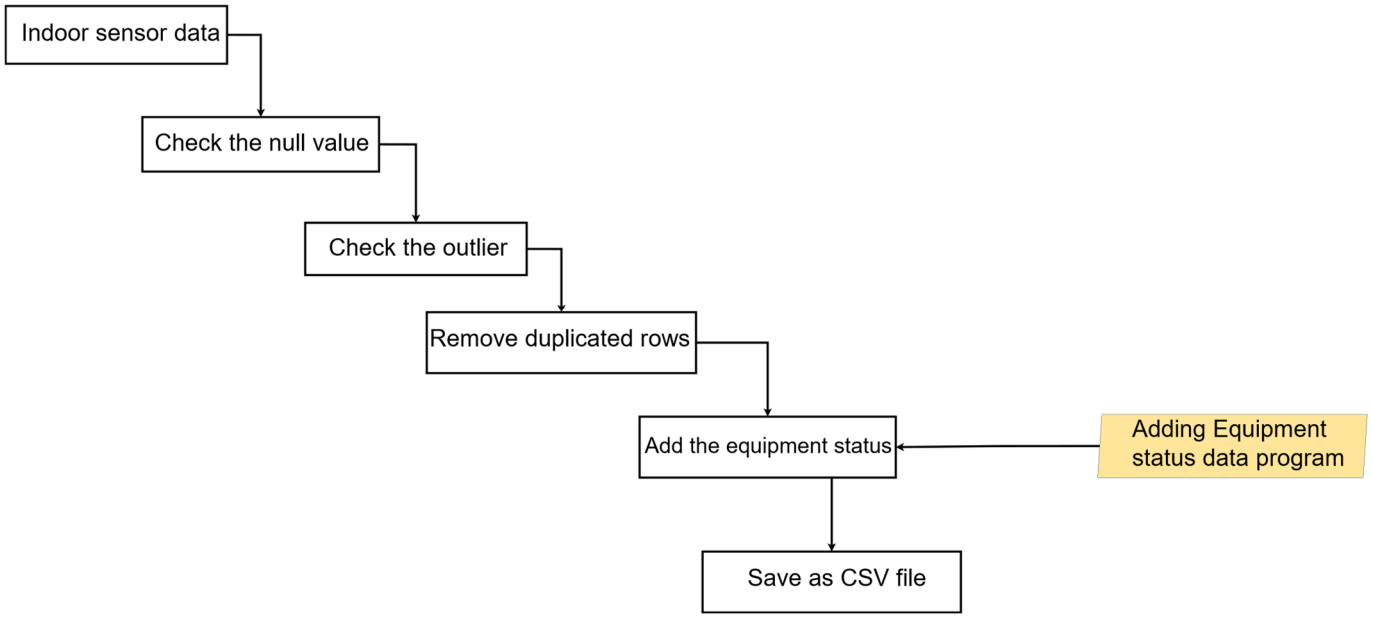


Figure 5. Data preprocessing flow diagram.

After standardizing the data, we transformed the data again to the original 2D, then saved the scaler mean and standard deviation of the standard scaler for the model deployment. We transformed data in slide window format, including batch size, step size, and features for all three split datasets. The sliding window function was implemented with Python—with a format of 30 sequences for the step size and 30 for the forecast temperature value. Next, the data were transformed to tensor format because the model was implemented with the Pytorch framework for a faster training time and provided the facility for implementation and model deployments. Before training the model, we reset the model and set all hyperparameters as the hidden vector set 4 with a learning rate of 0.001. The MSE loss was used for the loss function, and the Adam optimizer was used for the optimizer. After that, the model and all the data were converted to a GPU consumable format to train the model. Then, the model saved the best loss value for every step. After obtaining the best loss value, we stopped the model training. In order to evaluate and plot the model performance, we loaded the best loss value model; then converted data back into CPU to plot with the matplotlib library of Python, since that library does not support GPU;

forecasted the temperature value using the validation set; and then plotted the value. Then, the value was evaluated using two evaluation matrices, the RMSE (Root Mean Square Error) in Formula (13) and the R^2 (coefficient of determination) in Formula (14).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\gamma_i - y_i)^2} \tag{13}$$

where,

N = number of observations;
γ_i = actual value for the ith observation;
y_i = predicted value for the ith observation.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \gamma_i)^2}{\sum_{i=1}^{N}(y_i - \beta_i)^2} \tag{14}$$

where,

β_i = mean of the actual values.

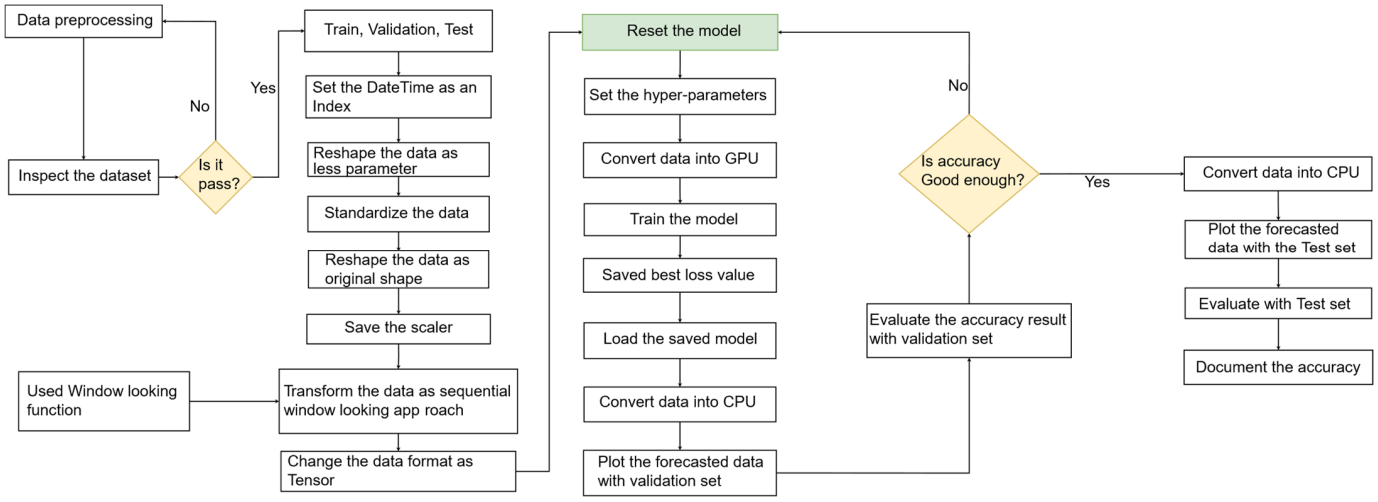


Figure 6. Model training process flow.

RMSE provided the absolute measure of the prediction error magnitude; the value gives the range 0 to 1 and a result value close to 0 represents a good performance accuracy, whereas a value close to 1 represents decreased accuracy. R^2 relates inversely to the previous matrix; a value close to 1 has a better performance accuracy and a value of 0 is given otherwise. After determining two of these evaluation matrices with the validation set, if the result was still not good enough, we stepped back to reset the model and conduct all the steps mentioned above; this step is considered state-of-the-art in machine learning. After trying several times, the result was satisfied; then we moved to the next step, using an unseen test set as the final stage evaluation for the model. That stage was the same as the validation evaluation steps, and finally, all accuracy performance was documented for later use.

The designed mm-LSTM model implementation with Pytorch frameworks like those mentioned above, especially the capability of faster model training and other support for the model. The model was trained on an AMD Ryzen 5 2600 six-core processor, RAM 64 GB, GPU NVIDIA GeForce RTX 3080 specified PC.

2.5. System Architecture of Open-Ventilated Greenhouse Equipment Control

Using the system architecture described in Figure 7, the open-ventilated greenhouse equipment control system primarily included three main parts: the server side, sensor data side, and web automation (client side). Sensor data were fetched through API every 40 s,

and the server requested the equipment status data by using an HTTP request to the web automation side as soon as sensor data were received. That step was required due to the model input parameters, including the fan and top spray operation status. Then, the model passed to the data preprocessing step. The whole process in this stage was the same as the mentioned data preprocessing step before model training in Section 2.4. The next section handles the inconsistent sensor data after preprocessing.

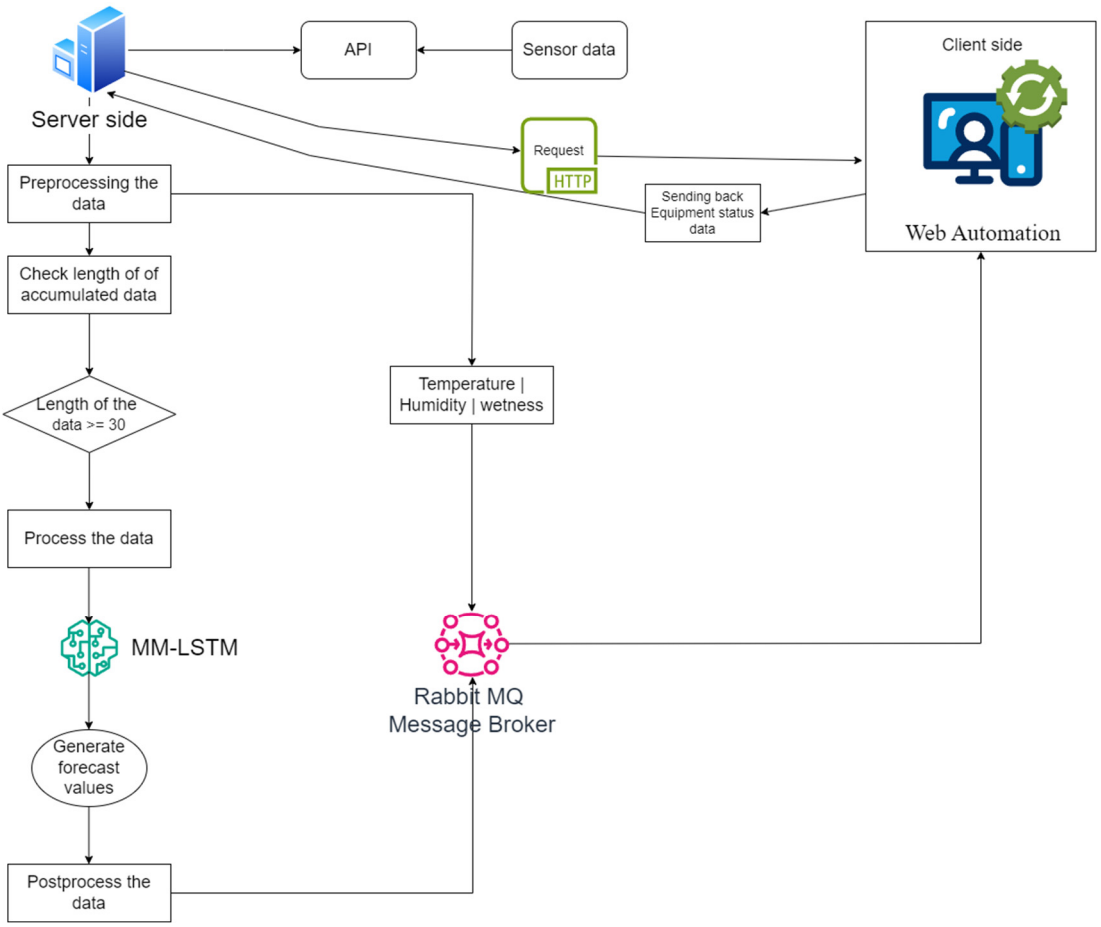


Figure 7. Overview system architecture of open-ventilated greenhouse equipment control via web integration with a cooperative data-driven approach.

Handling the inconsistent sensor prioritized two main factors: first, preprocessed solid data points should accumulate 30 data points as fast as possible, and second, the accumulated solid data points should be at most 30 data points. These two things were necessary for this system because obtaining faster forecasting values from the model input reduces the unfavorable effect on the microclimate of the greenhouse because the climate can dramatically switch from presently sunny to cloudy in the next 30 min, so obtaining consecutive forecasting values sequentially was necessary for this developed system. In addition, 30 data points for model input are considered not to have a long sensor data gap of accumulated data point between, which means if 40 accumulated solid data points are received at that time the system will only take the last 30 data points of 40, so the first 10 were wasteful for waiting time and not applicable.

This inconsistent data handling process is designed as in the flow diagram above (Figure 8). Four queue conditions are used. First, the raw data queue accumulates 40 data points. Those sensor data, which pass to the preprocessing step to check the condition of the number of solid data, remain solid if the data are within the values of 18 to 22, leading the process to execute Queue 2, which will wait for another 20 raw sensor data points, and then recheck the number of solid data points that additionally remained; if the preprocess

data points remained solid data points between 23 and 26, the process will wait for only 15 more raw data points; or if the solid data points received were greater than or equal to 27 after the preprocessing of the raw data, the process will only wait for an additional 5 raw data points. That way, inconsistent data are handled carefully and aligned with our prioritized conditions. After the accumulated solid data points received were equal to or greater than 30 data points in that stage, the data processing step was followed.

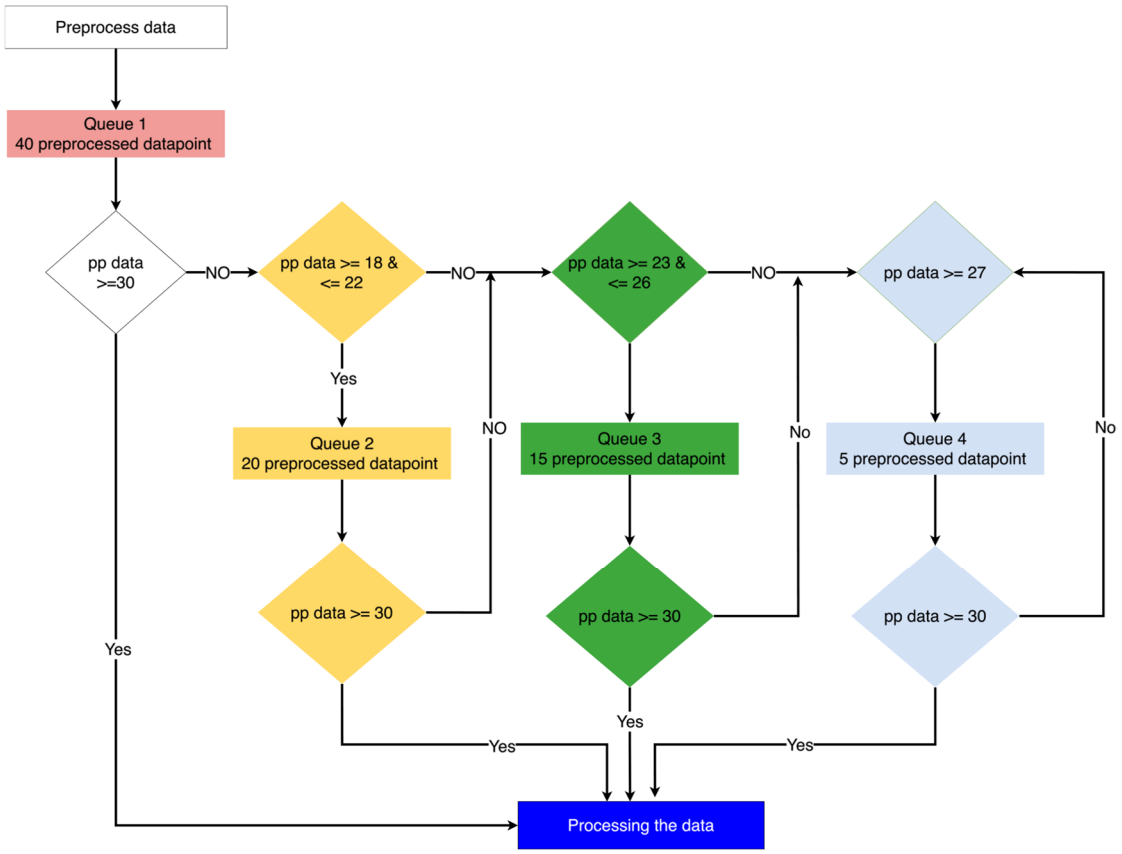


Figure 8. Inconsistent sensor data handling process flow.

The data processing step included two main things, including generating the forecasting timestamp of 30 data points, which represents the past 30 data points. We generated 30 data points and also had to attach the representant timestamp to each data point. In that step, generating 30 timestamps that represented the past 30 data points required an additional method to handle this process, since receiving 30 data points was inconsistent in time intervals. Hence, knowing the representative timestamp for the forecasting value was not simple. Secondly, the data during the model deployment stage were quite different from the model training phase, since upcoming sensor data might have a different distribution curve than the training phase as is the nature of environmental data, so we used another standardized method called the partial fit method to handle data standardization in the model deployment phase, described in more detail in the data standardization section.

This process flow (Figure 9) represents the handling of the inconsistent intervals of past data to forecast timestamps. There are a total of 4 steps: First, the time difference interval of 30 collected data points was extracted and temporarily kept as a list; second, the last timestamp from 30 data points was extracted to create a new forecast timestamp as a first timestamp. Third, the rest of the 29 timestamp intervals were added from a temporary keep list as a way of knowing the interval of the previous data point; and finally, 30 forecast timestamps were generated representing each past data point.

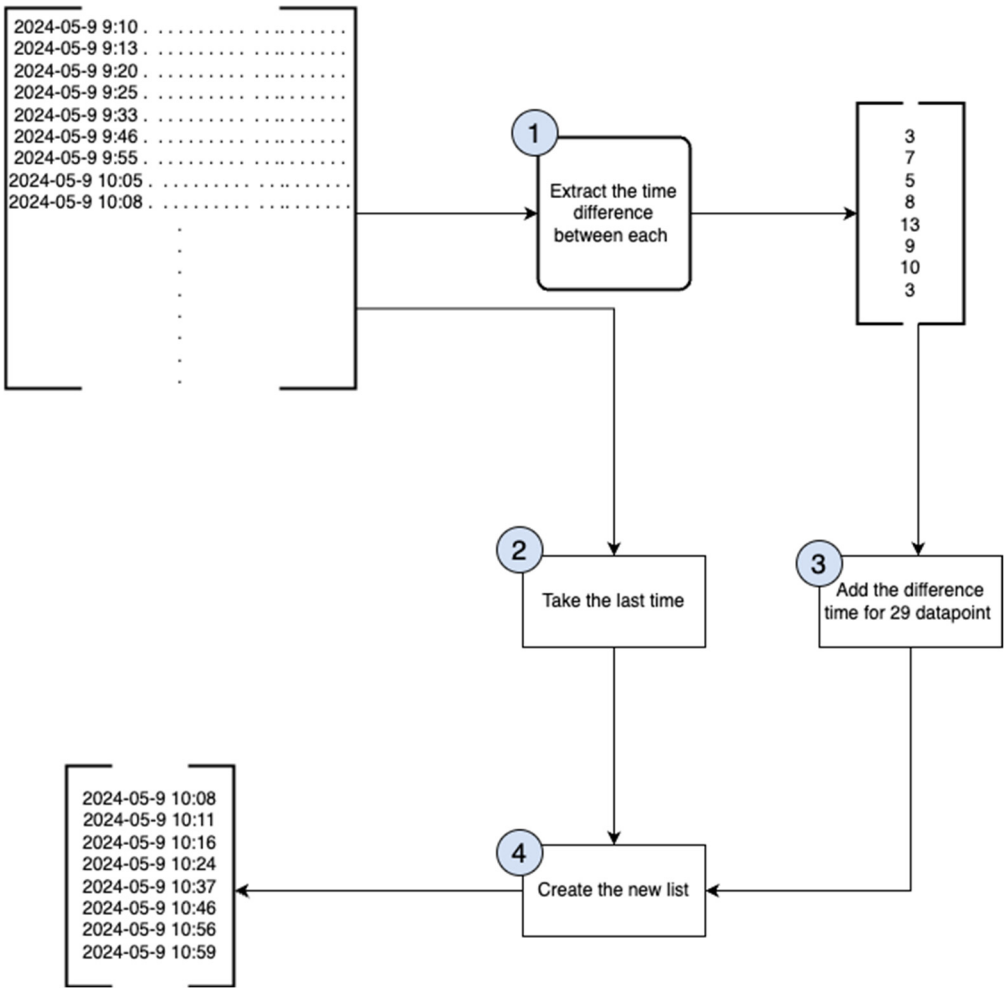


Figure 9. Generating forecasting timestamp that represented the past timestamp process flow.

Data standardization flow (Figure 10) described the first saved scaler value during the training stage, which was loaded. Second, the deep copy method was applied to keep the original mean and standard deviation values in memory to avoid interrupting. Then, the partial fit method [26] was used to update the mean and standard deviation values for upcoming new data if required. However, the value did not update directly through the original scaler values. Rather, it was separately kept as a new scalar and then concatenated with the original and a new scaler to transform the data. Of course, in the post-processing stage, data will be reconverted to the original value using the inverse transform method.

Model deployment was just one part of our system that saved the model with the best-lost value operating in the model development stage. The model needed to run in GPU in the same way as during the training time; during system operation, the model feeds 30 solid data points with 6 features and generates 30 forecast values that are attached with a timestamp, and 30 forecast data points are represented over approximately 1 h; however, that period depends on the incoming sensor data interval since it is inconsistent. In the post-processing stage, the Rabbit MQ message broker sends data to the client's web automation side with a cooperative data-driven approach, which means the raw sensor data and forecasting value. The raw sensor data temperature, humidity, and leaf wetness values are sent to the client's web automation side every 40 s through the message broker. This was required because equipment control did not solely rely on the forecasting value.

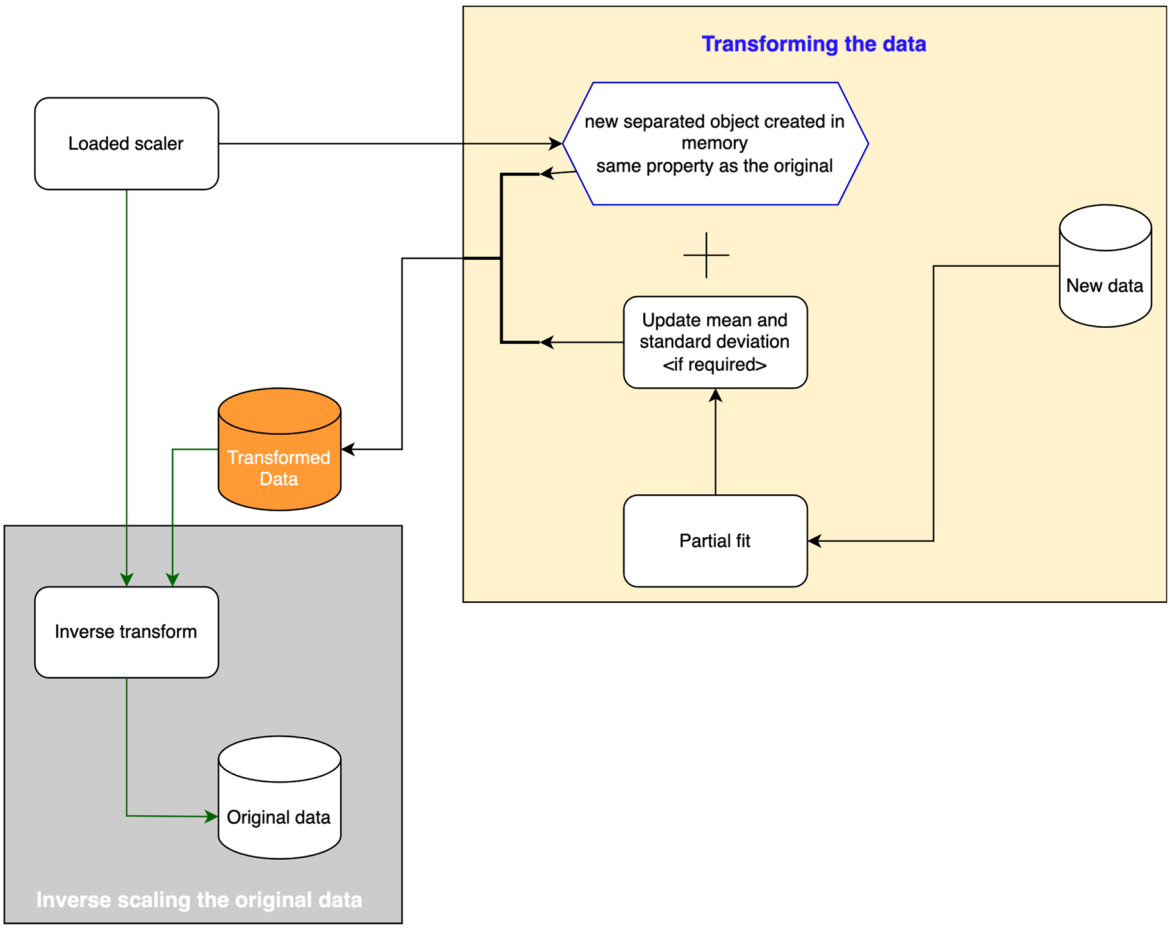


Figure 10. Data standardization flow through model deployment.

2.5.1. Web Automation

The web automation used the Selenium open-source tool via a greenhouse web portal that existed before. This tool was primarily intended to test the web pages with the different types of service tools provided. However, that tool had a more comprehensive efficiency and could be applied in multiple scenarios. Selenium has three different types: the Selenium web driver, one intended to automate web browsing automatically by programmatically supporting various program languages; the Selenium Grid designed to use and test web platforms or webpages on multiple drives, browsers, and operating systems that can run in parallel; and Selenium IDE which offers the record and playback tool for creating and running test cases of web pages without requiring programs. In our condition, web integration used the Selenium web driver with the Chrome browser by controlling the HTML element with Xpath and finding the element that triggers the button clicking and menu selection as per requirements.

Several problems were faced during the initial operation using the Selenium web driver, where running on a Chrome web driver with our system for 3 to 4 h, the RAM consumption of the host PC became high, then the browser crashed after that. After analysis, the main issue lay on the web driver, which needed to reset every time, so the problem was solved by refreshing the Selenium web driver for every function call; RAM overload was not raised throughout the experimental days. Another organization has already developed this web portal for our system and just used that web portal for the purpose of adaptive equipment control.

2.5.2. Equipment Control Condition in Web Automation

This diagram (Figure 11) describes the initial equipment operation condition when the web automation's forecasting value is generated. First, check the forecast temperature list's first and middle index values (denoted as 1 in the diagram). The temperature threshold is set at 35 °C, the humidity threshold is set at 68%, and the leaf wetness threshold is set at 500. If the first and middle index values are higher than the temperature threshold, perform the next step; otherwise, check the second condition (denoted as 2 in the diagram), which has a 12 index value and 27 index value, and compare those two values to the temperature threshold. If either of those conditions is true, perform the next step. In the next step, the fan will operate from the start time to the end, corresponding to the forecasting timestamp duration. Additional conditions are checked in the 0.5 top and side spray; these will only operate at a humidity of the sensor less than the humidity threshold and humidity, which is equal to 10. This extra check of all equipment conditions is necessary because the raw sensor data are not past preprocessing, which might include outlier values. If the temperature exceeds the temperature threshold plus 1.5 °C and both the humidity and temperature conditions are true, the 0.5 mm top and side spray will turn on. Likewise, the 0.3 mm top spray checks both conditions for a humidity less than the humidity threshold and a temperature greater than the temperature threshold plus 1 °C, which was considered an additional 1 °C because the vaporization effect of the 0.3 mm foggy was less than the 0.5 mm mist spray. If both conditions are true, the 0.3 mm top spray will operate.

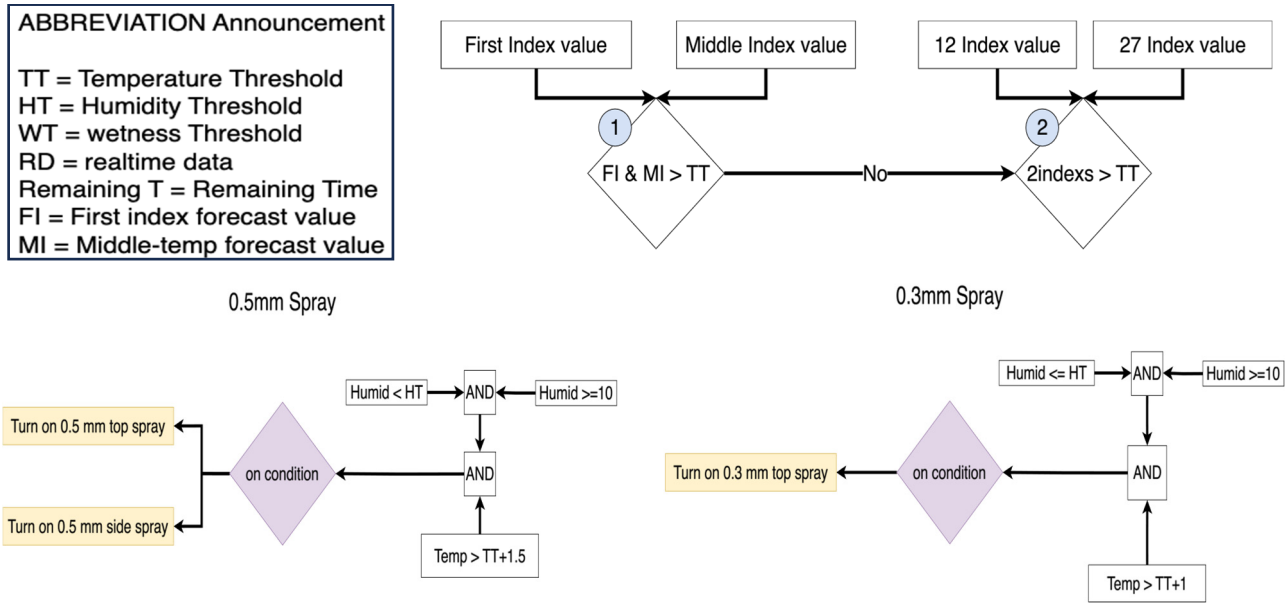


Figure 11. Initial equipment control for a fan, 0.5 mm top and side spray, and 0.3 mm top spray.

After initial operation for both fans, the system exhibits 0.5 mm mist and 0.3 mm foggy on and off conditions. In the next step, the extra fan condition (Figure 12a) is considered to be on either at a temperature higher than the temperature threshold plus 3 °C or at a humidity greater than the humidity threshold plus 2%. It will turn off either under a humidity less than or equal to the humidity threshold or a temperature less than or equal to the temperature threshold or an operation remaining time of less than 4 min (Formula (15)).

$$R_t = D_i - E_t \quad (15)$$

where,

R_t = remaining time, D_i = initial duration, E_t = elapsed time.

$$D_i = \frac{T_l - T_f}{60} \quad (16)$$

where,

T_l = last timestamp, T_f = first timestamp.

$$E_t = \frac{T_p - T_f}{60} \quad (17)$$

where,

T_p = present time.

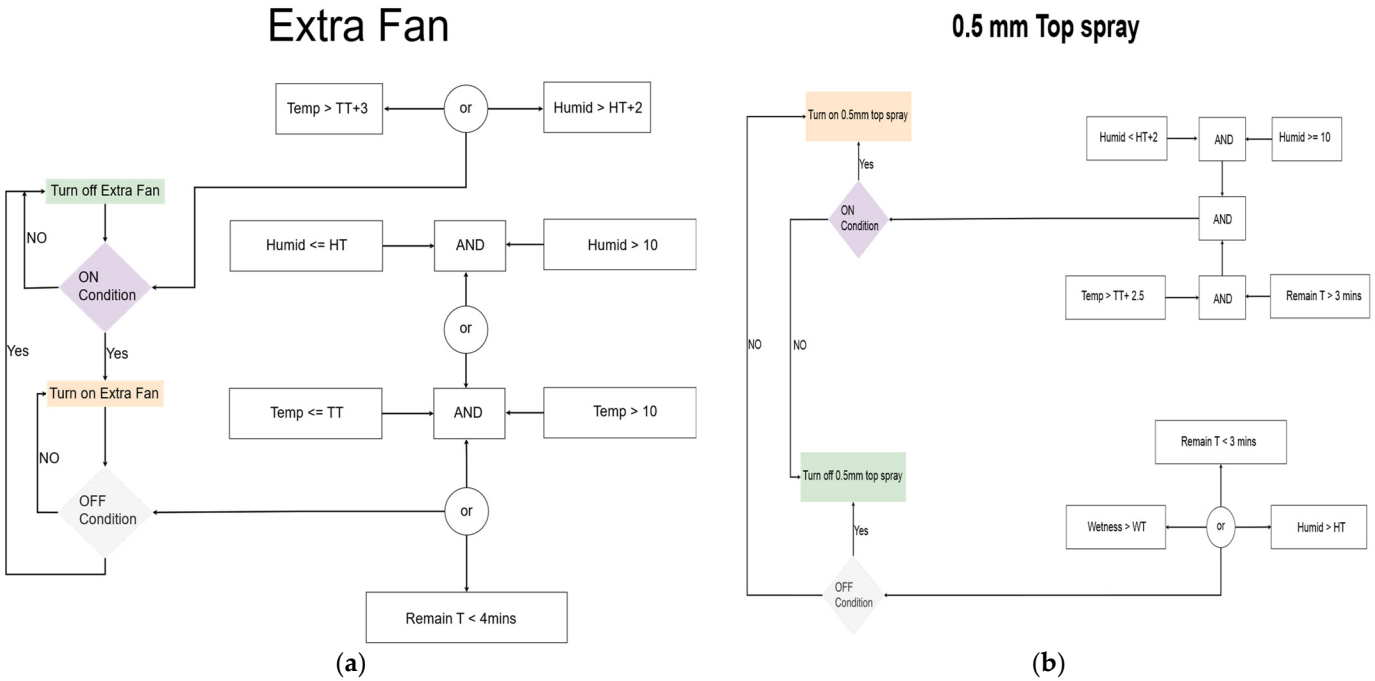


Figure 12. On and off condition of an extra fan (**a**) and 0.5 mm top spray (**b**).

The 0.5 mm top spray condition (Figure 12b) will start operating if the humidity is less than the humidity threshold plus 2% and the temperature is greater than the temperature threshold minus 2.5 °C. Moreover, the remaining time should be greater than 3 min. That spray will turn off if either the leaf wetness value is greater than the wetness threshold or the humidity is greater than the humidity threshold, or if the remaining time is less than 3 min.

The 0.5 mm side spray (Figure 13a) will start operating either at a temperature greater than the temperature threshold plus 1.5 °C or a humidity less than the humidity threshold or a leaf wetness value less than the leaf wetness threshold and a wetness value greater than or equal to 10 or a remaining time duration greater than 2 min. The spray operation will go off either under a temperature less than the temperature threshold plus 1.5 °C or a humidity greater than the humidity threshold or a remaining time of less than 2 min.

A 0.3 mm top spray condition (Figure 13b) will start operating either at a leaf wetness less than the wetness threshold or a humidity less than the humidity threshold and a temperature greater than the temperature threshold plus 1.8 °C and a remaining time greater than 2 min. It will turn off either under a temperature less than the temperature threshold or a humidity greater than the humidity threshold or a remaining time less than 1.5 min or a wetness value greater than the wetness threshold. Above all the conditions mentioned, "and" referred to both conditions needing to be valid for operating the equipment, whereas "or" referred to it being able to operate in an either/or condition. All equipment's on and off control conditions are included in the web automation side.

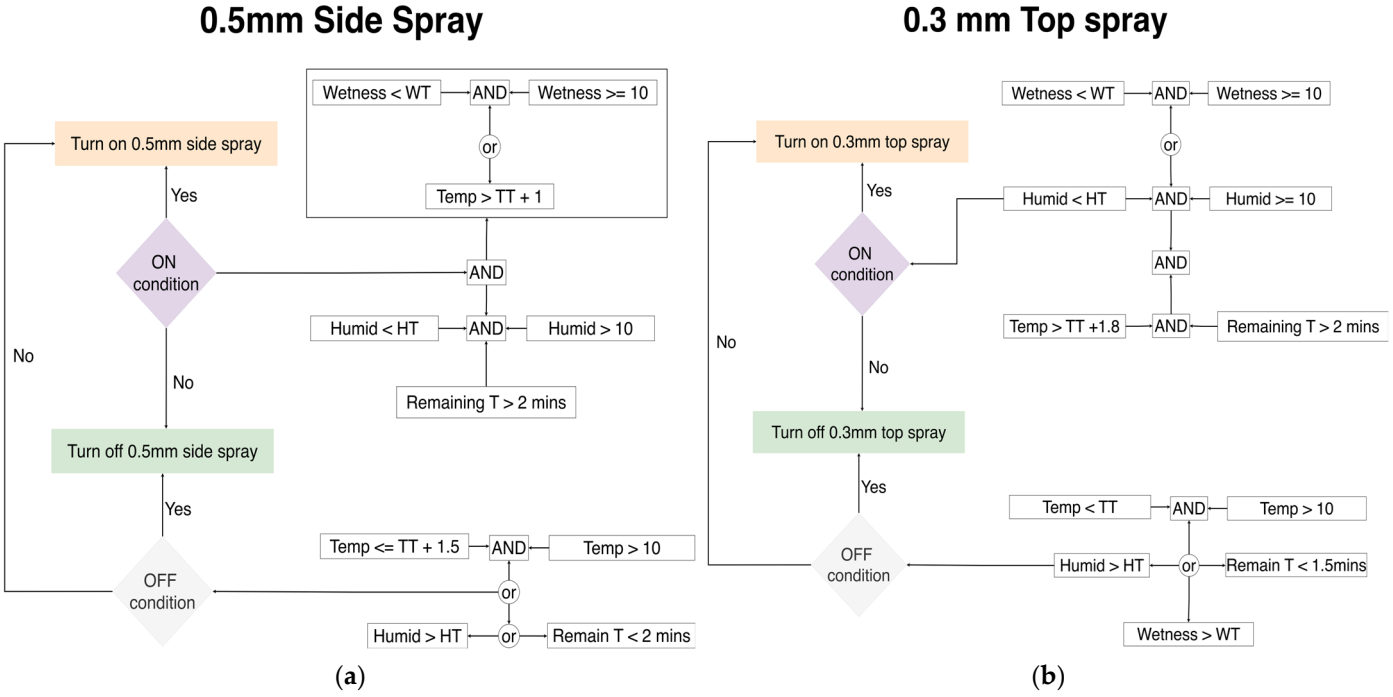


Figure 13. On and off condition of 0.5 mm side spray (**a**) and 0.3 mm top spray (**b**).

3. Results

3.1. Experiment Results on Indoor Enviromental Parameters

The experiment was conducted in the open-ventilated greenhouse for eight days, from 4 May 2024 to 11 May 2024. The deployed system operated for 24 h, paused for around 30 min for data analysis at midnight, and then resumed operation throughout the experimental days.

The eight experimental days' plots (Figure 14) described the hourly temperature comparison throughout the equipment day using the developed system for equipment control. The horizontal line of the plot represents the hour for 24 h, and the vertical line represents the temperature in degrees Celsius. The critical hour orange horizontal straight line refers to the approximated equipment operation that starts when the indoor temperature is higher than the temperature threshold (orange dashed line) according to the forecasting value generated by the mm-LSTM model. In the first 3 days, the outdoor temperature reached around 40 °C and over; then, the system could cool down to 34.5 to 36 °C most of the time during critical hours. The gray dashed line refers to the upper-temperature resistant threshold of 36 °C, which should not exceed that line for an ideal temperature. Equipment operation was not conducted if lower indoor temperatures were not required because of rain or other conditions, which are not necessary for cooling down. That means that the system and forecast value were ensured so that the equipment could not operate unnecessarily, which can waste energy sources and operation costs. However, there were a few times the systems were interrupted for the first few hours in the day, like the morning part of 8 May and the afternoon of 10 May, when the exceeded indoor temperature was higher than the outdoor because, like equipment off conditions, the indoor temperature was close to the outdoor temperature or higher than the outdoor. Equipment operation was interrupted due to a delay in sensor data accumulation. Even though sensor data were handled for inconsistency, at that time, the sensor did not send new update values and lagged, so generating the forecast value was delayed; that was not the system's fault. In addition to analyzing the microclimate, for which it would not be enough to look at temperature, humidity was also considered as an essential aspect for plants.

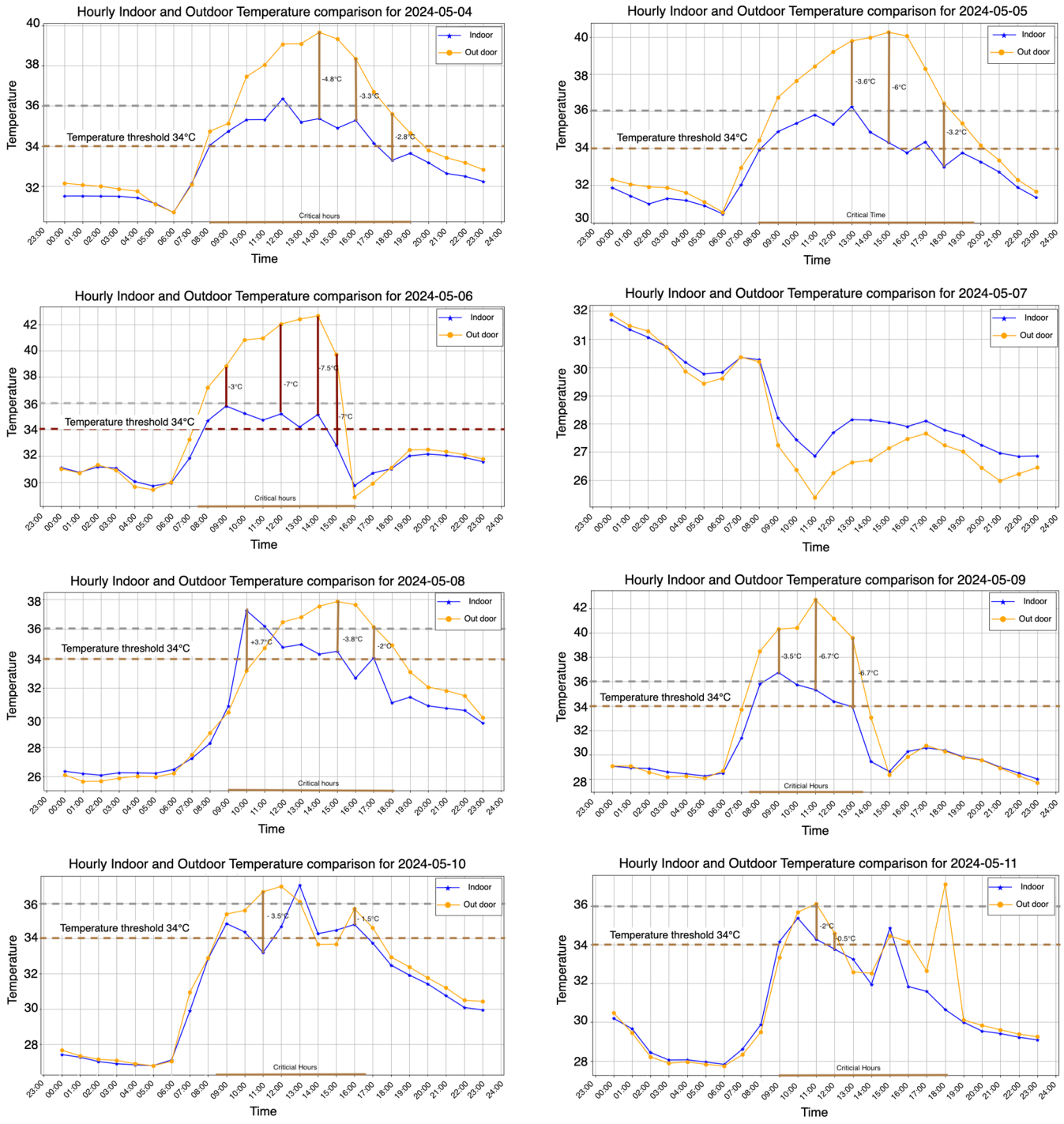


Figure 14. Hourly temperature comparison of the indoor and outdoor plot from 4 May 2024 to 11 May 2024.

One condition for humidity was that it could only be optimized during the equipment operation hours' duration in a day since the greenhouse did not have a dehumidifier or any other specialized equipment capable of controlling the humidity the rest of the time. Thus, humidity was higher in the morning and nighttime than in the critical hours (orange straight line) of daytime since the temperature and relative humidity correction were inversely correlated [27]. However, using mist and fog to cool down the temperature causes a higher relative humidity, so it is required to consider that condition. Throughout our experiment mentioned in Figure 15, the hourly humidity comparison plot, the humidity threshold (orange dash line) was set at 68%, an acceptable level for plants. Countries like

Thailand have a high humidity and hot temperatures that can easily affect the equipment during operation, so controlling the relative humidity was one of the tasks required in our system, and relative humidity achieved values of not much more than 68 to 70% the majority of the time during critical operation hours throughout the experimental days; the gray dashed line in the humidity plot represents the upper resistant threshold 75% which should not be more than that level for ideal humidity, which led to our system being able to adjust the mist and foggy operation favorable to the humidity threshold.

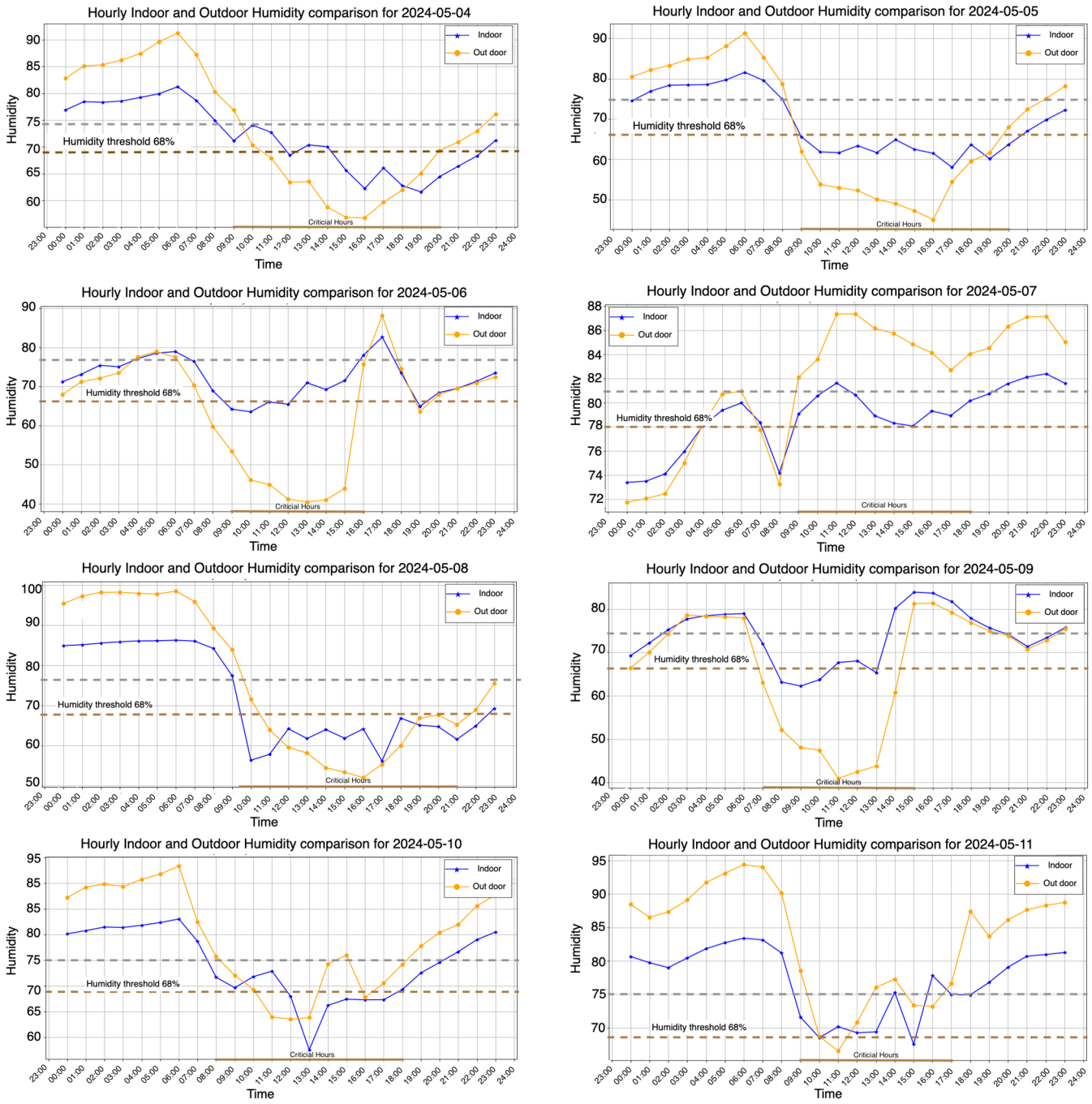


Figure 15. Hourly humidity comparison of the indoor and outdoor plots from 4 May 2024 to 11 May 2024.

One crucial fact was that the relative humidity could not precisely determine the actual effect of vaporization. That came in the leaf wetness value; it can detect and measure plant water vaporization levels in the greenhouse. The three plots (Figure 16) describe 30-min

intervals of leaf wetness throughout the experimental period; the horizontal axis of the plot represents the date and time, and the vertical axis represents the leaf wetness value. Apart from the first few days of wetness in some minutes, the value jumped to a threshold and reached 600 to 1200 at once; the rest of the time, it remained stable at 200 and 230 below the threshold. This wetness value demonstrated that our deployed system was not affecting the plant vaporization, which can cause fungus and disease in the long run. A stable vaporization level below the threshold can be seen in the plots.

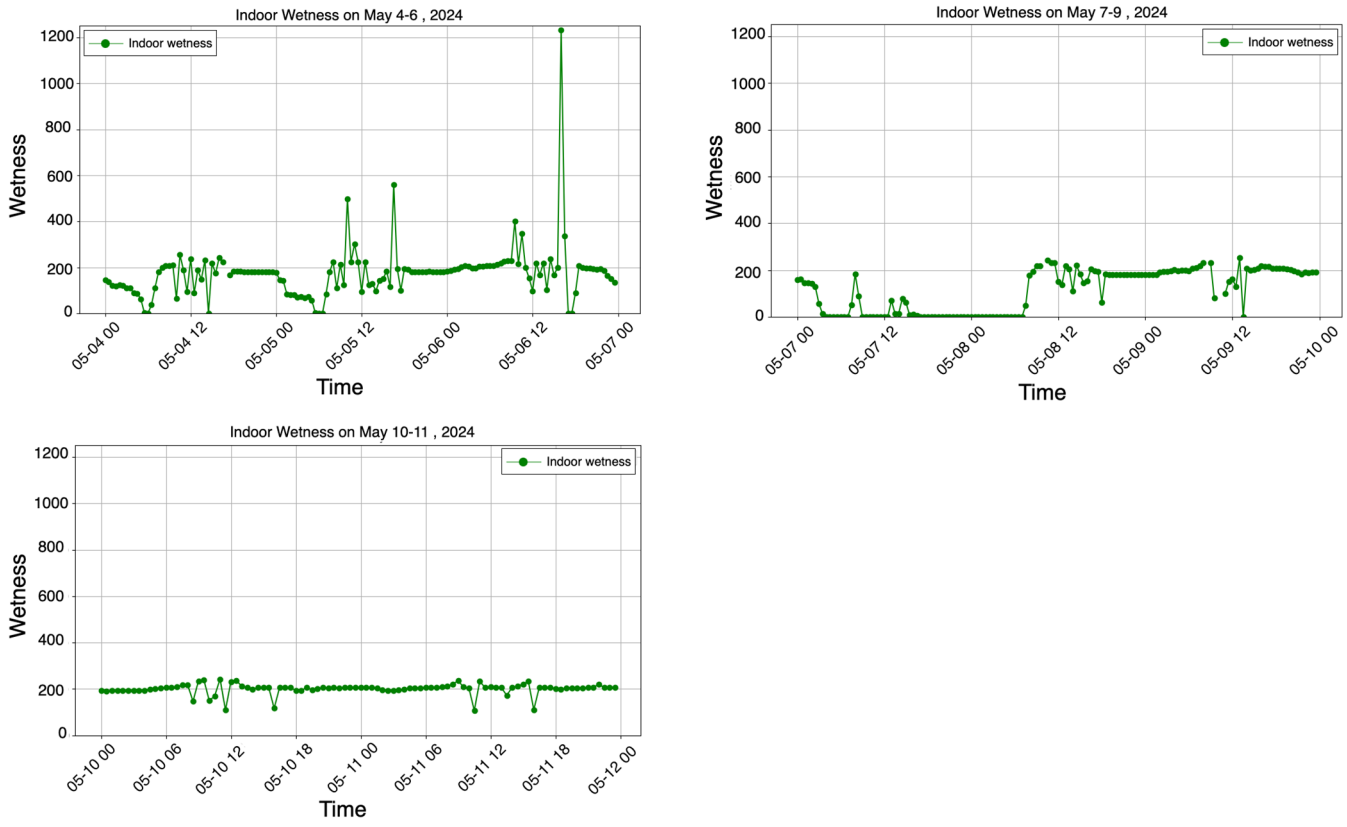


Figure 16. The 30-min interval leaf wetness value plot throughout the experiment from 4 May 2024 to 11 May 2024.

3.2. Model Performance Matrics

This mm-LSTM model was also compared with the baseline mm-LSTM, which only included one full content layer. The baseline model training processes and hyperparameters were the same as those of the designed mm-LSTM model. Our mm-LSTM model had a better performance accuracy than the based mm-LSTM in both evaluation matrices, RMSE and R^2 (Table 1). The model was trained again on all the data combined with the 47,270 datasets because much of the data could be used with a better accuracy on experimental days after the model was successfully evaluated with the test set.

Table 1. Model evaluation comparison result of the designed mm-LSTM model and baseline mm-LSTM model.

Name of the Model	RMSE	R^2
Designed mm-LSTM	0.49	0.976
Baseline mm-LSTM	1.38	0.90

The designed mm-LSTM model was re-evaluated on experimental days (Figure 17) since it had unseen data. This evaluation of the model's performance achieved RMSE

(0.515) and R^2 (0.976), which shows that the model was robust in capturing the dynamic pattern of environmental parameters like temperature.

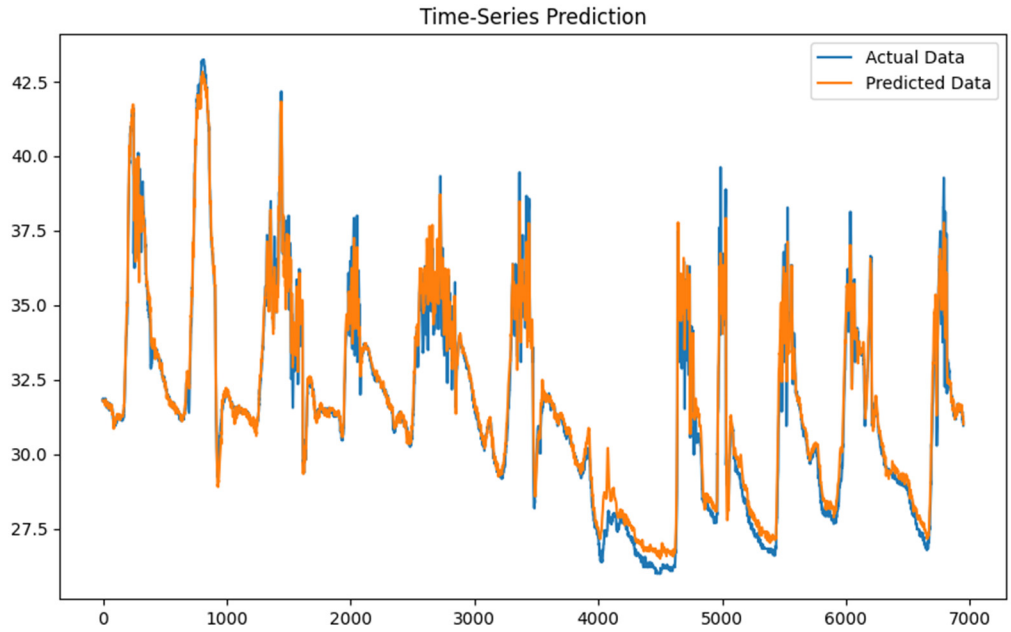


Figure 17. Model evaluation plot for experimental days; the x-axis represents temperature.

4. Discussion

This short experiment resulted in an effect on plants that grew in that greenhouse, which was only a visual difference from the before and after greenness of the plant, and it could not determine the plant's improvement in detail without future analysis. However, that was one of the factors in our deployed system's capability of adaptive controlling the open-ventilated greenhouse equipment for the plant ideal microclimate. The mm-LSTM model also could deal with temperature forecasting for approximately over 1 h. The abovementioned indoor environmental parameters demonstrated the adaptive equipment control that operated the implemented system throughout the experiment; the temperature can be maintained at 34.5 to 36 °C during the outdoor, either at 40 or over 40 °C. The main reason temperature reduction fluctuated throughout the system operation was that adaptive equipment control had to alternate between temperature reduction and humidity level; this means that too much temperature reduction in a one-time sequence can cause higher humidity levels.

On the other hand, the experiment conducted in the greenhouse was not equipped with any special equipment capable of decreasing the vaporization level that can reduce the humidity level, e.g., a dehumidifier. That is why reducing the temperature was not only the task of the developed system but also handling humidity levels that were not too far over the threshold. Promisingly, throughout the experiment, the humidity was below 70% most of the time.

One condition about humidity (relative humidity) could not describe the vaporization effect of plants, since the 0.5 mm and 0.3 mm spray, both top and side, used in our scenario can be prone to the over-vaporization effect and cause fungus and diseases of plants; consequently, there was a low yield rate at harvest time as well as the grower not being able to distribute to high-quality markets. Leaf wetness sensors can mitigate those issues with more precision detection on the plant level rather than the proximity of the whole greenhouse humidity. The leaf wetness sensor value mentioned above shows that the level of vaporization on the plant surface was not higher than the threshold of 500 most of the time. However, the deployed system had a few errors raised throughout the experiment due to sensor data delay, not the system itself. That also gives us more awareness of the de-

veloped system's need for consistent sensor data, and this kind of machine-learning model requires a consistent and frequent data supply. However, inconsistent sensor data were operated by other organizations; the developed system was able to adapt to inconsistent data handling processes throughout the experiment. Obtaining consistent sensor data for the machine-learning model was the most crucial thing; good data can lead the model in a promising way.

Due to the time limitation of this research, the plantation was not considered a complete crop cycle, and only a ventilated greenhouse setting was focused on in this research; however, it might be possible to use this proposed system in another ventilation type of greenhouse. As mentioned earlier in the relative humidity result analysis section, humidity control was concerned only with the equipment operation period in the daytime (critical hour). However, due to the more precise moisturization detection at the plant leaf level, this proposed mm-LSTM model and developed system architecture could use practical greenhouse settings with specific plant types.

The motivation for this research is also part of reviewing previous research findings, since open-ventilated greenhouses had a gap in equipment control by a data-driven approach, and actual experiments were required in countries like Southeast Asia. As with the nature of indoor greenhouse plantations, the effect of external environmental factors like weather conditions and methodological patterns could not be neglected, especially in open-ventilated greenhouses rather than fully enclosed greenhouses. Research studies conducted in Indonesia [12] and similar open-ventilated greenhouses also equipped with water spray (mist system) that control with a timer mechanism for 5 min on and 10 min off during critical times (daytime) were able to reduce the temperature to approximately 33 °C and humidity to 85% during their experiment in an actual greenhouse setting. That showed that even the temperature reached the optimum level regardless of the higher humidity level. One crucial parameter, humidity, could not be ignored, as mentioned, and a higher precision level will be required for controlling over-vaporization, which was not considered in their experiment. Considering the precision of the vaporization level, a Romanian type of greenhouse with no motioned ventilation [14] was experimented on with temperature and humidity control with the method of operation of fans, curtains, and water spray to control the temperature and humidity and equipped with a leaf wetness sensor to mitigate the risk of plant disease. However, that research did not use machine-learning forecasting techniques to operate the system. A two-day experiment was conducted using a researcher-implemented actors' control system. The result on temperature was good concerning that location's climate condition. However, it might not be possible to apply this technique due to insufficient information about the greenhouse ventilation type and different external factors like the weather.

The data-driven robust model predictive control (DDRMPC) framework was proposed for greenhouse climate control, such as temperature, humidity, and CO_2 concentration levels [13]. This method was supported by support vector clustering with a weighted generalized kernel. It demonstrated a better result of a 14% reduction in operation cost than a 4% rule-based control and less violation in temperature constraints. Experimenting with simulations rather than an actual greenhouse and a lack of detail on the greenhouse ventilation type created a gap for research on open-ventilated greenhouse settings and tropical climate patterns in terms of the optimization of adaptive equipment controls. This conducted research filled that gap, carrying out actual experiments in tropical regions' open-ventilated greenhouses with growers able to obtain an ideal level of temperature and water vaporization with consideration of adaptive equipment control with the purposed system and the mm-LSTM model being usable in countries like Thailand or other Southeast Asian nations with a reasonable setup cost for greenhouses. One advantage of the developed system is that it could leverage the fully automated control system by using this developed web integration and sensor data cooperation in a semi-automated web portal controllable greenhouse, which stood up in this research in the smart greenhouse use case.

5. Conclusions

The proposed architecture fetches the sensor data using API every 40 s on the server side. After the sensor data are sent to web automation (to control the equipment), the server side requests equipment status data using an HTTP request. Raw sensor data (temperature, humidity, leaf wetness) are also sent to the web automation side via Rabbit MQ, which is intended to cooperate with the sensor data and forecasting value in a later stage. Raw data are passed to the data preprocessing stage to filter out the outlier, null value, and duplicated value, and then inconsistent sensor data with different sequence levels are handled. The techniques prioritize two conditions: accumulate a solid 30 data points as fast as possible, and these data points should be at most 30. The next step includes forecasting 30 timestamps representing the previous 30 inconsistent data points. Also, the data are standardized using a method called partial fit, which can handle outliers of the mean and standard deviation that cooperate with the loaded scaler saved during model training. The deployed model generates 30 temperature forecast values attached to the represented timestamp generated in the previous step. That forecast is sent to the web automation side via Rabbit MQ for equipment control. The web automation side uses a selenium web driver that triggers the clicking of the HTML element by xpath in the greenhouse web portal.

An experiment was conducted for 8 days using the deployed system in an open-ventilated greenhouse, which had been mentioned before, and the temperature and humidity were analyzed. The temperature was maintained at 34.5 to 36 °C during outside temperatures of approximately 40 °C. Humidity did not increase much during the equipment operation hours (daytime), mostly 68% to 70%. One condition for humidity was controlling the humidity, which could only be performed during the equipment operating time because the greenhouse did not have specialized equipment to control the humidity for the rest of the time. The leaf wetness value, which can more precisely determine the vaporization level than humidity, was analyzed and achieved no more than 300 most of the time throughout the 8-day experiment. The model was also re-evaluated with experimental days' data that resulted in an RMSE of 0.515 and an R^2 of 0.97.

This described the model as robust in capturing environmental parameters like temperature and able to obtain a good accuracy performance even on unseen datasets throughout the experimental days. The temperature and leaf wetness value showed that controlling the equipment in the open-ventilated greenhouse was one of the essential things to do for plants, as well as obtaining a reasonable yield rate and quality of agri-food. The developed architecture of equipment control was also capable of operating for 24 h without interruption apart from the faced sensor data delay. The mm-LSTM model also fits with this kind of time-series data forecasting. Through the experimental days, smart greenhouses required maintenance aspects for equipment that should be considered for medium- or large-sized farms. Less human interaction can cause an insufficient water level and quality as well as spray nozzle clog, so instilling more sensors for the water level, water pressure, and quality monitoring was a necessary step and supported machine-learning techniques for the maintenance aspect to be considered for future research in open-ventilated smart greenhouses.

Author Contributions: Conceptualization, K.M.M.T. and T.H.; methodology, K.M.M.T.; software, K.M.M.T.; validation, K.M.M.T. and T.H.; formal analysis, K.M.M.T.; resources, K.M.M.T., T.H. and T.P.; data curation, K.M.M.T.; writing—original draft preparation, K.M.M.T.; writing—review and editing, K.M.M.T. and T.H.; visualization, K.M.M.T. and T.H.; supervision, T.H. and T.P.; project administration, K.M.M.T. and T.P.; funding acquisition, T.H. and T.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research is financially supported by Thailand Advanced Institute of Science and Technology (TAIST), the National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology, Sirindhorn International Institute of Technology (SIIT), and Thammasat University (TU) under the TAIST Tokyo Tech Program.

Data Availability Statement: Collected data used in model training and system deployment are already mentioned in the original analysis results in the Results Section. Corresponding authors can be contacted with further inquiries.

Acknowledgments: The authors are grateful for agriculture knowledge and support from Ornprapa Thepsilvisut from the Department of Agriculture in the Faculty of Science and Technology, and the academic advisor Teerayut Horanont from SIIT, and support in the greenhouse experiment from Teera Phatrapornnant from Nectec and Opas Trithaveesak in providing the sensor data. This research is partially supported by the Center of Excellence in Digital Earth and Emerging Technology (CoE: DEET), Thammasat University.

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

References

1. Abid, H.; Ketata, A.; Lajnef, M.; Chiboub, H.; Driss, Z. Impact of greenhouse roof height on microclimate and agricultural practices: CFD and experimental investigations. *J. Therm. Anal. Calorim.* **2024**, *149*, 5483–5495. [CrossRef]
2. Nemali, K. History of controlled environment horticulture: Greenhouses. *HortScience* **2022**, *57*, 239–246. [CrossRef]
3. Calise, F.; Cappiello, F.L.; Cimmino, L.; Vicidomini, M. Dynamic modelling and Energy, economic, and environmental analysis of a greenhouse supplied by renewable sources. *Appl. Sci.* **2023**, *13*, 6584. [CrossRef]
4. Sapounas, A.; Katsoulas, N.; Slager, B.; Bezemer, R.; Lelieveld, C. Design, control, and performance aspects of semi-closed greenhouses. *Agronomy* **2020**, *10*, 1739. [CrossRef]
5. Riudavets, J.; Moerman, E.; Vila, E. Implementation of integrated pest and disease management in greenhouses: From research to the consumer. *Integr. Pest Dis. Manag. Greenh. Crops* **2020**, *14*, 457–485. [CrossRef]
6. Benni, S.; Tassinari, P.; Bonora, F.; Barbaresi, A.; Torreggiani, D. Efficacy of greenhouse natural ventilation: Environmental monitoring and CFD simulations of a study case. *Energy Build.* **2016**, *125*, 276–286. [CrossRef]
7. Jung, D.-H.; Kim, H.S.; Jhin, C.; Kim, H.-J.; Park, S.H. Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse. *Comput. Electron. Agric.* **2020**, *173*, 105402. [CrossRef]
8. Liu, Y.; Li, D.; Wan, S.; Wang, F.; Dou, W.; Xu, X.; Li, S.; Ma, R.; Qi, L. A long short-term memory-based model for greenhouse climate prediction. *Int. J. Intell. Syst.* **2021**, *37*, 135–151. [CrossRef]
9. Esparza-Gómez, J.M.; Guerrero-Osuna, H.A.; Ornelas-Vargas, G.; Luque-Vega, L.F. Deep Learning for Greenhouse Internal Temperature Forecast. *Int. J. Comb. Optim. Probl. Inform.* **2023**, *14*, 86–94.
10. Kang, M.; Weng, Y.; Pang, H.; Li, L.; Fan, X.-R.; Hua, J.; Chang, F.; Wang, X.; Ma, L. Semi-autonomous greenhouse environment control by combining expert knowledge and machine learning. In Proceedings of the 2020 Chinese Automation Congress, Shanghai, China, 6–8 November 2020; pp. 7500–7504. [CrossRef]
11. Fan, L.; Ji, Y.; Wu, G. Research on temperature prediction model in greenhouse based on improved SVR. *J. Phys. Conf. Ser.* **2021**, *1802*, 042001. [CrossRef]
12. Eduard, R.; Ruslan, W.; Iskandar, I.; Setyanto, D. Setting temperature and humidity with a misting system in a pilot greenhouse at CISAUK-Tangerang, Indonesia. *Appl. Sci.* **2022**, *12*, 9192. [CrossRef]
13. Chen, W.-H.; You, F. Semiclosed greenhouse climate control under uncertainty via machine learning and data-driven robust model predictive control. *IEEE Trans. Control. Syst. Technol.* **2022**, *30*, 1186–1197. [CrossRef]
14. Zarnescu, A.; Ungurelu, R.; Varzaru, G. Controlling the temperature and humidity in a greenhouse. In Proceedings of the 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), Cluj-Napoca, Romania, 23–26 October 2019; pp. 168–171. [CrossRef]
15. Hemming, S.; de Zwart, F.; Elings, A.; Righini, I.; Petropoulou, A. Remote control of greenhouse vegetable production with artificial intelligence—Greenhouse climate, irrigation, and crop production. *Sensors* **2019**, *19*, 1807. [CrossRef] [PubMed]
16. Jensen, M.H.; Malter, A.J. *Protected Agriculture: A Global Review*; World Bank: Washington, DC, USA, 1995.
17. Anand, G.; Rajeshkumar, K.C. Challenges and threats posed by plant pathogenic fungi on agricultural productivity and economy. *Fungal Biol.* **2022**, 483–493.
18. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
19. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
20. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the 29th International Conference on Neural Information Processing Systems (NIPS'15), Montreal, QC, Canada, 7–12 December 2015; pp. 802–810.
21. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; pp. 103–111.
22. Brownlee, J. *Long Short-Term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning*; Machine Learning Mastery; EBook; 2017.

23. Kuber, V.; Yadav, D.; Yadav, A.K. Univariate and Multivariate LSTM Model for Short-Term Stock Market Prediction. *arXiv* **2022**, arXiv:2205.06673. [CrossRef]
24. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef] [PubMed]
25. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *arXiv* **2011**, arXiv:1201.0490. [CrossRef]
27. Wedyan, G.; Farah, H.; Osama, T. Analysis of air temperature, relative humidity and evaporation over Iraq using ECMWF reanalysis. *Indian J. Ecol.* **2021**, *48*, 446–452.

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.