A Thesis for The Degree of Doctor of Philosophy

Development of Artificial Intelligence-based Climate Control System for Smart Greenhouse

인공지능 기반의 스마트 온실 환경 제어 시스템 개발

August 2020

Graduate School of College of Agriculture and Life Sciences
Seoul National University
Major in Biosystems Engineering

Dae-Hyun Jung

Development of Artificial Intelligence-based Climate Control System for Smart Greenhouse

Advisor: Hak-Jin Kim

Submitting a Ph.D. Dissertation of Biosystems Engineering
August 2020

Graduate School of College of Agriculture and Life Sciences
Seoul National University
Department of Biosystems Engineering
Dae-Hyun Jung

Confirming the Ph.D. Dissertation written by
Dae-Hyun Jung
July 2020

Chair	Joong Yong Rhee	(Seal)
Vice Chair	Hak-Jin Kim	(Seal)
Examiner	Young-Jun Park	(Seal)
Examiner	Jung Gyk Son	(Seal)
Examiner	Soo Hyun Park	(Seal)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Greenhouses are widely used to create an artificial environment that is favorable for the growth of plants. Maintaining a suitable temperature, humidity, and carbon dioxide concentration is essential for environmental control within a greenhouse because these factors affect plant development, quality, and production levels. Plants exposed to unusually low or high temperatures or humidity may experience stress, disease, and/or death, causing significant financial losses for growers. Greenhouse climate control must consider a complex and nonlinear system in which variables are highly dependent on external climatic conditions and greenhouse design. Therefore, establishing a method for the precise control of the internal greenhouse climate is vital for responding to these dynamic changes and achieving efficient climate management. Therefore, the objective of this study was to develop an improved artificial intelligence-based climate control system using big data from a smart farm.

In this study, a model employing PD-band- and P-band-based linear algorithms was proposed for the control of greenhouse actuators in the management of temperature, humidity, and CO_2 concentrations, and the performance of this model was verified. To optimize the settings for the PD-band model, response surface analysis and experimental analysis were conducted. Based on the results, the conditions required for optimal ventilation control were established, and the optimal values for each factor were applied to a real greenhouse temperature control system; the resulting root mean square error of 1.25 ℃ confirmed that the optimized coefficients improved climate management performance.

The deep-learning-based prediction model developed in this study was designed to respond to changes in the climatic conditions of the greenhouse due to operational changes. Artificial neural network, neural network autoregressive, and recurrent neural network with long short-term memory

architecture (RNN-LSTM) models were developed to determine the best approach to predicting changes in temperature, humidity, and CO_2, concentration. The RNN-LSTM exhibited the highest overall accuracy for temperature and CO_2 prediction (5% standard error of prediction and 0.81−0.96 R^2, respectively). Various training conditions were also analyzed, and the 5−30 min prediction performance was evaluated. Using a convolutional neural network (CNN) with LSTM, it was possible to predict environmental changes within actual greenhouses, and it exhibited a slightly stronger performance than did the RNN-LSTM. These results clearly demonstrate the potential for the use of deep-learning-based prediction models in greenhouse control.

A method of determining the optimal actuator signal using backtracking was also introduced to the structure of an output feedback neural network (OFNN). This was employed in calculating the costs derived from the target climate settings, the current climate values, and the predicted change in the climate values after 30 min, and an optimization method to reduce these costs was devised. Gradient descent, which is commonly employed in machine learning and deep learning research, was employed in the form of an OFNN. Using a multiwindow ventilation control experiment in the field, it was verified that the actuator signal was more sensitive to environmental changes than the existing linear algorithm. For energy conservation, the driving energy for the actuators was used as a function in the cost gate to consider the energy consumption when determining the control signal. The actual energy savings were subsequently confirmed in a field application.

The developed artificial intelligence-based climate control system was designed to minimize the input energy and errors associated with the set values for more efficient control decisions. This reduces the input energy and thus has positive economic implications that should encourage its adoption in smart farms in the near future. In addition, the increasing need for intelligent environmental control technology in other industries suggests that the system proposed in this study is of great significance for the horticultural industry in

general.

Keywords: Smart farm, Climate control, Climate predictive model, Intelligence control, Model predictive control,
Student Number: 2014-31170

1 INTRODUCTION

1.1 STUDY BACKGROUND

South Korea's agricultural industry continued to grow until the early 2000s. However, in recent years, the average farm income has stagnated, the self-sufficiency rate for grain has decreased, the size of the rural population has shrunk and aged, and climate change has made it more difficult to attain target crop yields, which has resulted in a lower growth rate for agriculture (Yoo et al., 2016). As the crop yield and production has become more erratic, and the price of agricultural products has fluctuated dramatically, and food security for the nation has become a concern. Total agricultural production drastically increased from KRW 32 trillion in 2001 to KRW 44 trillion in 2012 before leveling off until 2016. Despite this increase in overall production, the average yearly income per farm has remained relatively unchanged at around KRW 11.30 million between 2001 and 2015, while the size of the rural population significantly decreased from 4.0 million in 2000 to 2.6 million in 2015. In order to overcome these challenges, the South Korean government has implemented a new industrialization policy that promotes agriculture in connection with the primary, secondary, and tertiary industries (Shin et al., 2015).

Based on this new industrialization policy, the number of entrepreneurs launching startups increased from 752 in 2014 to 1,530 in 2016. In addition, governmentally operated industrialization service centers support the production and marketing of agricultural products in collaboration with related organizations, including agricultural product processing centers. As a result, tangible results in terms of an increase in sales have been attained. However, to more meaningfully advance Korean agriculture, it is necessary to actively embrace scientific technology, especially techniques associated with the 4th Industrial Revolution (e.g., the Internet of Things [IoT], big data, drones, robots, and artificial intelligence [AI]), for use in production, distribution, consumption, and upstream and downstream sectors. Indeed, smart agriculture systems have emerged in several developed countries, such as the U.S., the U.K., and Germany, based on advanced information technology and full agricultural mechanization. In terms of revenue, the North American region

dominated the market in 2016 and is expected to remain dominant in the near future.

Prospects for Smart Farming Technology

Smart farming involves the use of autonomous agricultural machines, intelligent robots, and autonomous farm control applications based on IoT, big data, and AI. Smart farms can solve current problems within the agricultural industry, such as labor shortages, decreasing productivity, and the stagnation of farm income. The global smart agriculture market was valued at USD 5.79 billion in 2016 and is expected to reach USD 18.22 billion by 2025, growing at a compound annual growth rate (CAGR) of 13.5% over the forecasted period (Figure 1). Despite this, South Korea's agricultural industry still operates according to the standards of the 3rd Industrial Revolution, where data are collected in field surveys, from statistics, or in a specific way, the information derived from modeling using the collected data is provided to farms, and then each farm manually adjusts its operations based on the information. Smart farming technology cannot be truly implemented until the technologies associated with the 4th Industrial Revolution are employed in downstream industries as well as in agricultural production, distribution, and consumption.

In addition, automatic management systems need to be more fully embraced. For example, when structured data obtained from IoT devices and unstructured data collected from SNS are analyzed using deep learning, AI, and semantic web technologies, the analysis results can be uploaded to a cloud system and used for autonomous agricultural machines, robots, and drones. The Netherlands, one of the most advanced countries in terms of smart farming, focuses on software development, including big data analysis, on the grounds that there is no significant difference in hardware technology such as greenhouse construction and sensors in less advanced countries (Lee et al., 2016).

Precision agriculture or satellite farming is a farming management concept based on measuring, observing, and responding to intra- and inter-field variability in crops (Ding et al., 2018; Roy et al., 2002). The goal is to provide a decision support system for farm management while preserving resources and optimizing the return on inputs. In precision agriculture, farmers collect real-time data on soil, weather, air quality,

crop maturity, and labor costs, which helps them to make informed decisions. In order to implement such systems in Korea and enhance the production of food, the agricultural industry requires a technology infusion. In particular, smart agriculture includes software and hardware components, with equipment requiring application software to be embedded and hardware requiring a connection to network services.

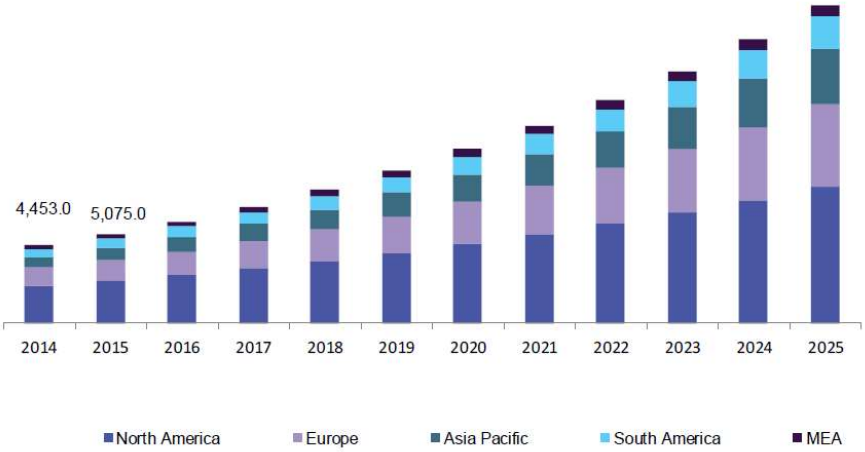

Figure 1. Smart agriculture market size and growth prospects (revenue in million USD) 2014−2025 (Sources: Agfunder, EPA, World Bank Reports, Drone Blog, ICT Update, company annual reports, primary interviews, and Grand View Research)

Artificial Intelligence (AI) in Agriculture

AI involves the development of computer systems that are capable of performing tasks that usually require human intelligence. Applications of AI in the agricultural industry include precision farming, livestock monitoring, drone analytics, agricultural robots, labor management, smart greenhouse management, soil management, and fish farming management. The increasing strain on the global food supply due to the growing population and government support for the adoption of modern agricultural techniques have been the key growth drivers for AI in agriculture. As such, the adoption of AI in agriculture adds significant value to a farm as a whole as well as to the consumer supply chain (Lee et al., 2016; Muangprathub et al., 2019). In fact, the overall use of AI in agriculture was valued at USD 835 million in 2019 and is expected to reach USD 4.0 billion by 2026, at a CAGR of 25.5% between 2020 and 2026.

AI in agriculture provides intelligent farming solutions for farmers that allow them to grow crops more efficiently through the analysis of real-time data for weather conditions, temperature, soil moisture, weed detection, crop health monitoring, and crop prices (Wolfert et al., 2017). Machine learning, computer vision, and predictive analytics technologies have thus been increasingly adopted in the agricultural industry, mainly for forecasting crop yields (Figure 2). Other major factors that have contributed to the growth of AI in the agricultural market are the rising use of robots and drones on farms, increasing demand for enhanced crop yields, and the growing adoption of AI technologies such as machine learning, computer vision, and predictive analytics by farmers and agribusinesses (King, 2017).

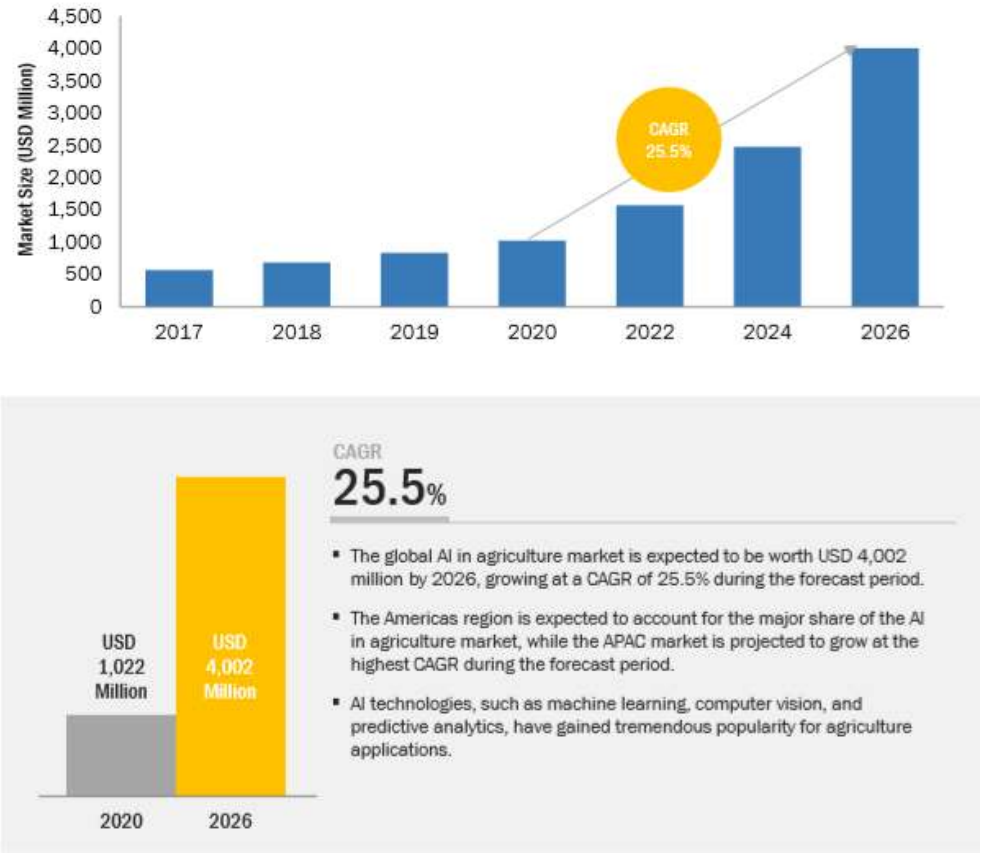

Figure 2. Global artificial intelligence (AI) market in agriculture, 2017–2026 (million USD) (top) and growth of AI in the agricultural market between 2020 and 2026 (bottom). (Sources: Annual reports, press releases, investor presentations, expert interviews, Association for the Advancement of Artificial Intelligence, Artificial Intelligence Association of India, Chinese Association for Artificial

Intelligence, Pattern Recognition and Machine Intelligence Association, and market analysis).

Machine-learning-enabled solutions are being increasingly adopted by agricultural organizations and farmers worldwide to enhance farm productivity and to gain a competitive edge in business operations (Park et al., 2016). In the coming years, the use of machine learning in various agricultural activities is expected to rise further. Technological advancements and the proliferation of machine-learning technology for farm data generation are some of the major driving factors for the use of AI in the agricultural market. The increasing use of deep learning algorithms in a variety of applications in agriculture is also a major driving force for the increasing demand for AI in agriculture (Kamilaris and Prenafeta-Boldú, 2018). Deep learning is a machine-learning technology based on multiple algorithms that explore relationships among data points. Deep learning typically uses artificial neural networks (ANNs) to learn patterns within multiple levels of unstructured data, including text, images, and sound.

Predictive analytics uses various techniques based on algorithms to analyze currently available data and predict crop yields, crop nutrient deficiencies, and plant health. Predictive analytics technology uses data from satellites and drones to virtually scout the area of a farm and to detect and classify specific anomalies. Farmers can then use this information to manage crop inputs and make agronomic decisions in order to improve overall productivity and crop efficiency (Morota et al., 2018; Wolfert et al., 2017).

The Future of Greenhouse Control system

Unlike outdoor cultivation, greenhouse-based cultivation has the advantage of allowing the specific control of the growing conditions. As a result, productivity and product quality can be improved by analyzing the growing conditions, including climate and crop cultivation parameters. For this reason, the number of greenhouses in a certain area tends to increase when precision control and air conditioning techniques are made available. The early form of cultivation under a structure was limited to the supply of water. However, technological development now allows the automation of the entire crop cultivation process.

The primary issues in greenhouse horticulture are increasing productivity, reducing costs, and increasing crop quality. Productivity can be improved using hybrid environmental control and management, spatial cultivation techniques, growth diagnosis and forecasting techniques, and remote expert services. Hybrid environmental control includes temperature and humidity control and carbon dioxide (CO_2) fertilization. Productivity can be greatly improved if relevant data are collected by IoT devices and analyzed using AI so that the environment can be automatically controlled. As such, a number of smart farms have installed a sensing system to monitor the growth of crops and an ICT system that transfers the information about crop growth and the internal climate and external weather collected from various sensors to a central server (Figure 3). As traditional agriculture focusing on the simple production of in-season crops is replaced by smart farming technology focusing on sophisticated control, crops can be produced irrespective of the season. Accordingly, more farms can adopt smart farming technology to increase their income, competitiveness, efficiency, and crop quality.

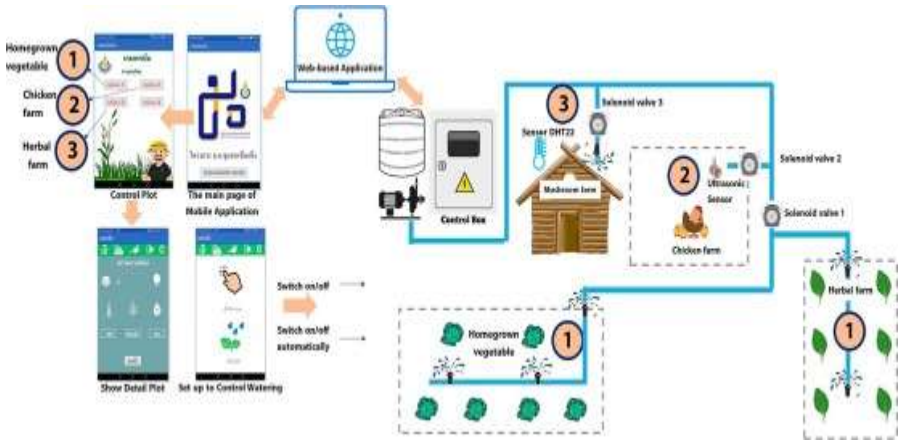

Figure 3. Overview of ICT-based smart farm

1.2. PROBLEM STATEMENT

A greenhouse artificially manipulates the environment so that it is suitable for the target crop. Of the indoor environmental conditions that need to be monitored, maintaining an optimal climate is of great importance because it has a significant

effect on the development, quality, and quantity of plants produced (Qian et al., 2015; Takahata and Miura, 2017). Over time, greenhouses have developed from very simple structures with crude control options to modern industrial structures that offer various ways to manipulate the environmental conditions. However, the automatic control of the greenhouse climate still requires further development. A greenhouse is a highly nonlinear and strongly coupled system that is heavily influenced by the external weather and the behavior of the actuators installed in the system (Zeng et al., 2012). In recent years, many studies have investigated greenhouse environmental control (Coelho et al., 2005; Fitz-Rodríguez et al., 2010; Piñón et al., 2005), but the systems proposed in these studies have not entered the commercialization stage. At present, the performance of commercially available controllers for greenhouses is very limited. For example, there are numerous problems with precise temperature control, as shown in Figure 4. It is common for the air to become overheated because of the delay in releasing heat from the greenhouse, while frequent operation causes fluctuations in the temperature. In order to improve environmental control, experience-based linear algorithms can be used, but they often make the system less convenient to use. This section will summarize some of the key issues that need to be resolved in greenhouse environment management.

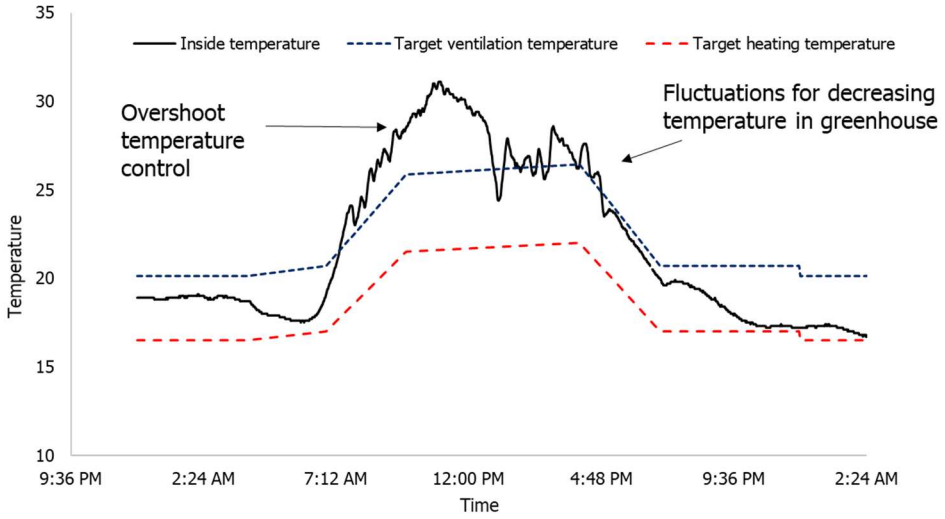

Figure 4. Example of temperature changes in a typical greenhouse using a commercial controller.

Nonlinear and coupled operation of environmental controllers

A greenhouse is a very complex nonlinear system that requires a meticulous approach to controlling the environment. Figure 5 presents the relationship between the physiological and growth phenomena observable in a greenhouse and the system elements that can be controlled. For example, ventilation is associated with both the external and internal environment of the greenhouse, affecting the temperature, relative humidity, and CO_2 concentration within the greenhouse, which in turn affects the atmospheric pressure and ultimately evapotranspiration, sap flow, and crop respiration. In addition, the amount of light inside a greenhouse can be adjusted by installing artificial lights and/or shading curtains. Light intensity is a very important environmental factor because it directly affects thermal radiation and photosynthesis within the greenhouse. CO_2 enrichment also has a very complex impact on crop physiology, being used as the main raw material for the photosynthesis and anabolism of crops. Typically, CO_2 from the external environment is used, but CO_2 levels can be controlled by injecting more during the day when photosynthesis is highly active. However, CO_2 concentration is very difficult to model because it is related to crop respiration. Nutrition and irrigation are control elements that affect the root zone of crops, which is involved in evapotranspiration and CO_2/O_2 exchange and affects nutrient uptake and crop growth. In addition, fogging and heating systems control the temperature and humidity inside the greenhouse.

Currently, commercialized controllers operate using set values for each influential factor based on a certain linear band as selected by the user. The difference in the performance of different environmental controllers depends on how precisely the empirical equation based on these coefficients can be calculated. This set-up has led farmers to prefer foreign-manufactured controllers in a domestic environment where commercial environmental controller technology and products are relatively insufficient.

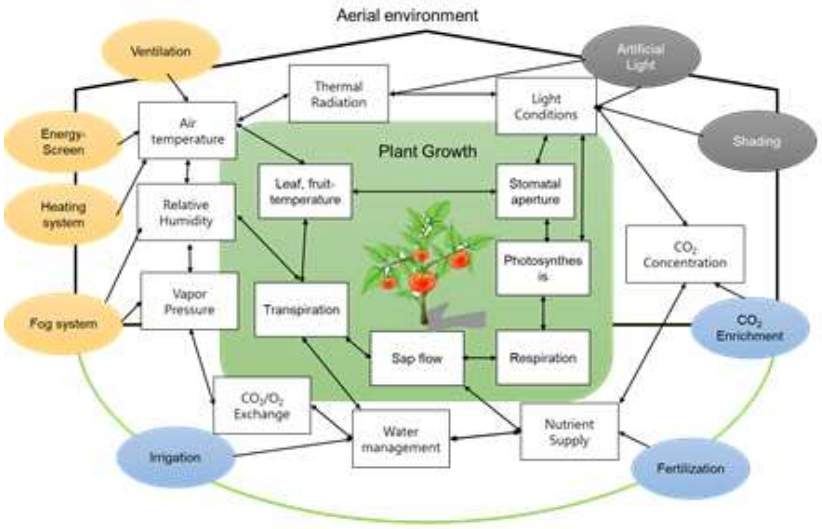

Figure 5. The relationship between various environmental control elements and the corresponding environmental factors within a greenhouse.

Complexity of environmental controller setting factors

The performance of an environment controller depends on the set values, which are often complex and sophisticated. Figure 6 displays the complex configuration screen for a commercial product. It is necessary to present each reference value when inputting the coefficients into the settings for the greenhouse, and it is necessary to tune it according to the structure of the greenhouse and the growth of the target crop. However, in most cases, this function is not used effectively on Korean smart farms.

Countries with advanced agricultural industries, particularly those in Europe, employ their own systems to produce various crops, thus increasing the quality and productivity and reducing costs. On the other hand, South Korean farms depend on imported equipment and lack appropriate cultivation and environmental management techniques. As a result, the crop production per unit area in South Korea is only half that of the Netherlands, which is a leading country for smart farms in Europe. Due to the costs involved, the majority of farms consider affordability rather than performance when purchasing smart farm equipment. However, because the core components of these systems are imported, when a failure occurs, immediate servicing is often impossible, thus damaging the farm's operations. Thus, the development of high-performance controllers that utilize domestic components is

urgently required, as is the establishment of Korean smart farm techniques that are suitable for the local agricultural environment.

Currently, domestic technology in this area is limited to IoT-based data collection and analysis, while temperature/humidity and heating/cooling systems continue to be manually operated. Thus, the majority of advanced domestic smart farms use imported multivariable environmental control systems that directly influence crop yield and quality. However, these imported control systems consider neither the weather conditions nor the growing conditions for crops in South Korea. For this reason, farmers have difficulty setting the controls (Figure 6). Although farmers need to be trained in the use of the control software, their control efficiency has not improved as much as expected because of the lack of knowledge from the farmer. Because linear control models have some limitations in terms of environmental control, precise control technology that utilizes environmental big data from farms is required.

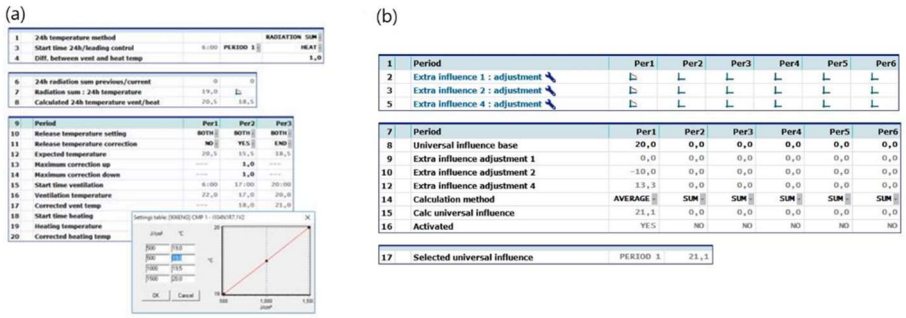

Figure 6. Examples of the setting screen for a greenhouse environmental controller: (a) the setting of influential parameters and (b) the setting screen for greenhouse ventilation control.

Uncertainty and lack of standardization in the use of AI

With the emergence of the 4th Industrial Revolution, various industries have harnessed AI to promote innovative productivity and increase efficiency. Smart farms are a noticeable example of this trend. In South Korea, many studies have actively sought to combine smart farm and AI technologies. In fact, deep-learning-based image analysis technology has been so actively developed that it has spilled over from agriculture into other industrial areas. In terms of automatic environmental

control systems for smart greenhouses, various attempts have been made to apply AI-based algorithms, but significant success has yet to be achieved.

Excessive energy use for environmental control

The high energy costs associated with the use of the actuators required for the ventilation, heating, humidification, and cooling systems within a smart greenhouse increase the overall cost of production for a farm (Maher et al., 2016). It is thus necessary to maintain an environment that optimizes crop growth while still considering the economic efficiency of the farm.

1.3. OBJECTIVES AND AIMS

1.3.1. OVERALL OBJECTIVE

This study aims to develop an AI-based climate control system that harnesses big data from a smart farm. This system will include various algorithms for the automatic control of the greenhouse climate. The results of this study are expected to facilitate the production of high-quality agricultural goods by minimizing labor costs and providing an optimal environment for the growth of crops in smart greenhouses.

1.3.2. SPECIFIC OBJECTIVES

The specific AI-based environmental control concept proposed in this study is shown in Figure 7 and Figure 8. In the first step, a set-point controller configuration based on a conventional linear algorithm is installed and tested. The second step is carried out based on the big data obtained from the linear controller, with various AI-based prediction models for environmental changes and control methods developed and tests. Finally, in the third step, an intelligent controller and a model that optimizes the control signals are proposed. These steps all focus on the control of rhizosphere nutrients and the environmental temperature and humidity. The overall goal is to develop a model that controls both factors using a sophisticated environmental controller.

In summary, this study proposes a smart farm environmental control technology

that employs various AI algorithms. The specific objectives are to

- Establish linear-based control logic for use in the actuators that maintain the climatic conditions of a greenhouse
- Develop machine-learning- and deep-learning-based algorithms for climate prediction modeling of various IoT/ICT-based environmental monitoring systems
- Create an AI optimization algorithm that sets the control signals based on the developed predictive model, and
- Assess the AI optimization algorithm based on the energy efficiency of the actuators during operation.

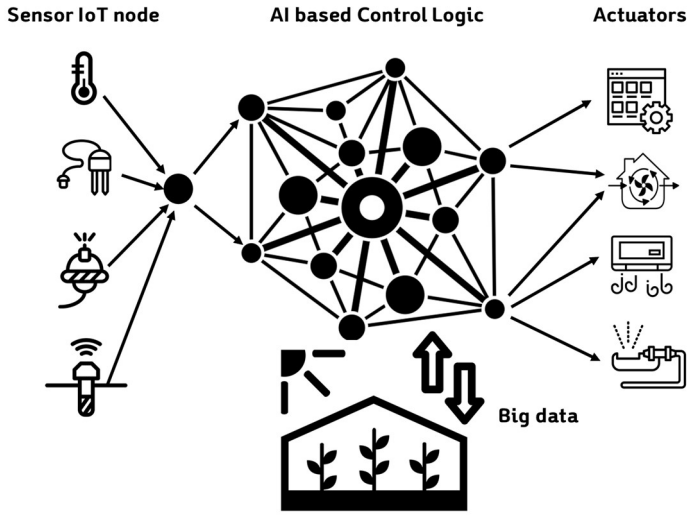

Figure 7. Development of a smart farm climate control system for greenhouses based on an AI prediction model.

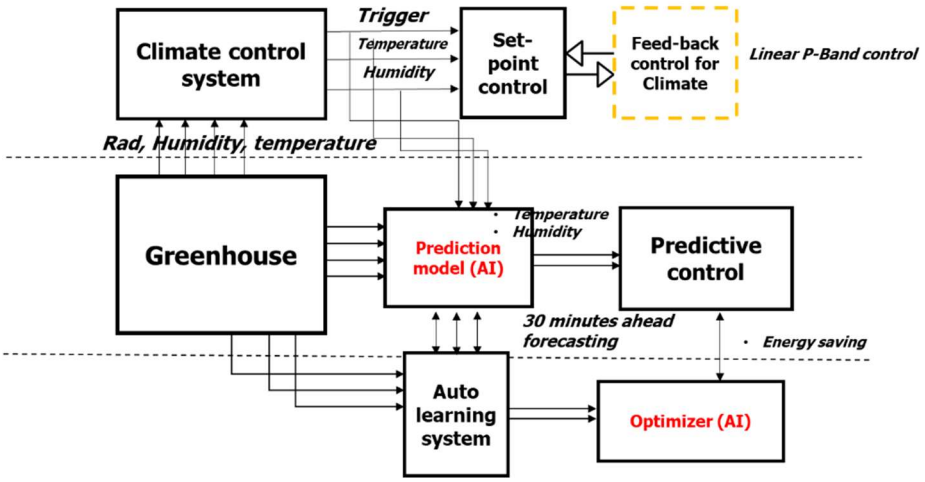

Figure 8. The specific AI-based environmental control concept proposed in this study.

Chapter 3 presents the development of the improved linear algorithms and explains the construction of a smart greenhouse, while Chapter 4 describes the construction of a climate prediction model using AI and proposes a prediction algorithm. Chapter 5 establishes a control optimization method that moderates the climatic conditions of a greenhouse based on the developed predictive model. Chapter 6 creates a modified algorithm that makes optimal control decisions and considers the energy efficiency of the actuators. The technologies discussed in Chapters 3–6 are then combined and the expected results for on-site scenarios are investigated. The expected contributions of this study are as shown in Figure.9.

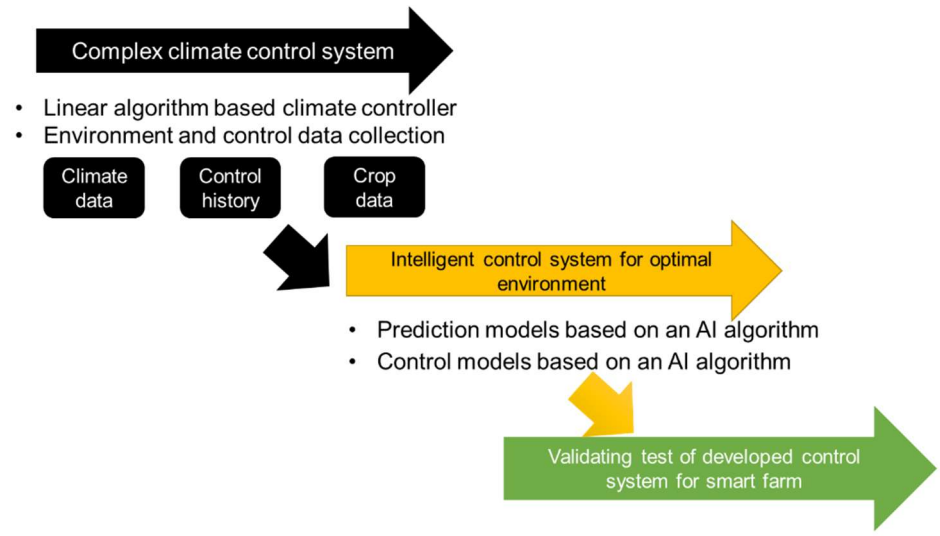

Figure 9. Step-by-step research outline for the present study in the development of second-generation smart farm technology.

2. LITERATURE REVIEW

2.1. OVERVIEW OF RESEARCH TREND

Smart farms have recently emerged as the future of agriculture and have started to receive active governmental support. In particular, technical development has been concentrated in the smart farm system. For example, smart farm technologies have been introduced to hobby- and leisure-based urban agriculture to create added value, meaning that ordinary citizens have become more interested in smart farms. There is a diverse range of smart farms depending on the farm size and design, crop, and cultivation method. Because various techniques need to be developed for different farm types, a variety of studies in many areas have been carried out (Yeo et al., 2016).

Table 1 presents the policy roadmap for smart farms established by the South Korean government. Government-initiated smart farms are now in a transition from second-generation to third-generation smart farms (Cha et al., 2016). AI technology is regarded as a key element in this phase. In second-generation smart farms, big data research has focused on platform development with governmental support. This groundwork intended to integrate data from various sensors and to standardize protocols. However, analysis engines that can optimally control the environment and enhance productivity have yet to be developed to an acceptable standard.

Table 1. Government strategic classification of Korean smart farms (Cha et al., 2016).

	1st generation smart farm	**2nd generation smart farm**	**3rd generation smart farm**
Data collection	Climate data	Climate and crop data	Climate and crop data Distribution and market data
Data analysis	Experience-based knowledge	AI algorithm	AI algorithm
Service	Automatic control	Automatic control	Automatic control

		Decision making	Decision making Crop growth management
Control system	Installed farm unit Control	Cloud control	Strategic crop harvest control by region

Currently, South Korean smart farms are restricted to importing main components (sensors, controllers, etc.) in order to install a control system. These farms face a large financial burden when purchasing, operating, and maintaining these systems. Moreover, because many system components lack compatibility, maintenance is very difficult and even minor failures often cannot be fixed. This lack of customer service forces some farms to abandon the use of these systems. In addition, the import-dependent structure of the South Korean smart farm market continues to release cultivation data from farms to the other conturies, where many system manufacturers and installers are located. Accordingly, these companies utilize the data for systems on domestic farms used to cultivate crops in greenhouses.

The technical development of environmental control systems in South Korea has not yet reached a stage where a farmer can understand how to apply the data (e.g., outdoor atmospheric temperature, wind speed, humidity, rainfall, etc.) collected by the minute by multivariable environmental control sensors in greenhouses to agricultural fields (Jeong et al., 2019). In addition, many farmers are not aware of the importance of growth data, which has to be manually recorded. Accordingly, it is difficult to recognize how the growth of crops is affected by the environmental control system in a smart farm facility. Finally, because multivariable climate control systems have unsatisfactory operability, precise control is difficult, and the efficiency and economic outcomes for the farm tend to be dependent on the grower's individual ability.

AI and big data technologies for agricultural use have been most actively developed in the US. Since 2014, an open-data policy has promoted the development of agricultural services. Recently, robotics has started to be applied to agriculture in conjunction with IoT. For example, Google utilizes AI and big data technologies in a system that monitors the moisture and growth of crops and assists in decision-making regarding the application of fertilizer and pesticide. In this system, ICT-

based sensors are installed in agricultural machines and fields to collect as much data as possible, the data are analyzed using big data methods, and optimal farming methods are proposed for each region. This system enables farmers to monitor soil conditions and weather forecasts and to control the growth of their crops.

In the Netherlands, 99% of greenhouses are glasshouses that are equipped with complex, precision environmental control systems, thus it is the leading country for smart farm technology. As a global leader in agricultural technology, it has already established the most efficient technology for crop production, developing various sensors and control solutions using the data that have been accumulated for decades and local know-how regarding the optimization of growing conditions. Based on agricultural ICT technology, production and quality have thus been optimized. Priva, a Dutch company, produces and exports the world's best control system for greenhouses.

The Netherlands is actively pursuing improvements in AI greenhouse control with Wageningen University & Research (WUR) as a leader in technology in agriculture (Hemming et al., 2019). The goal of a recent challenge along these lines was to remotely produce a cucumber crop within four months inside a greenhouse. The experimental greenhouse space and controls are provided by WUR, and the challenger teams were allowed to set up their own sensors and cameras. Each team extracted necessary data from the greenhouse and established their own ICT, model, and machine-learning algorithms (Figure 10) in order to decide on the control settings for the next day or period.

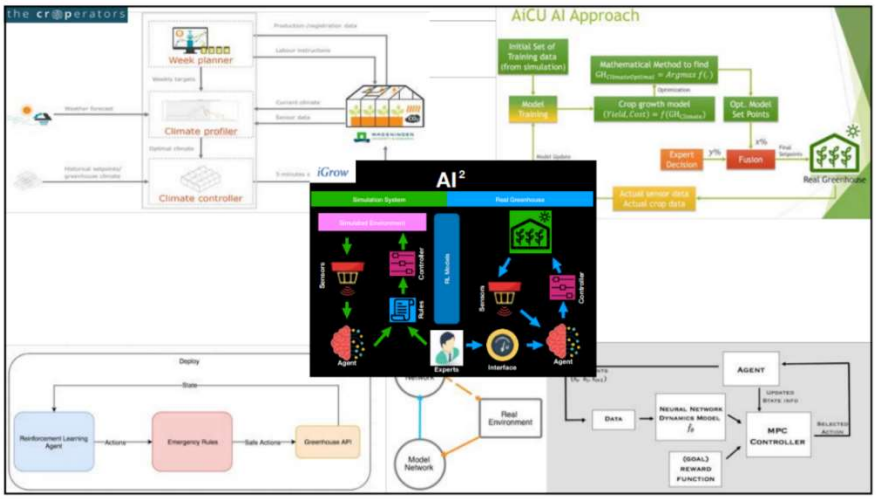

Figure 10. Concept art for AI-based control logic proposed for the autonomous greenhouse challenge hosted by Wageningen University & Research.

The world's leading countries in agriculture are actively introducing and utilizing smart farms in order to effectively solve labor shortages and other socio-structural problems faced by conventional agriculture. South Korea has also established active measures to develop and realize smart farms using ICT and AI technology. However, there is a gap in technology between South Korea and the leading countries, including the US and the Netherlands. Although Japan has undergone similar socio-structural changes to South Korea in some respects, it also belongs to the leading group. In Japan, many companies provide various ICT-based services, including tracking the management of crops, monitoring the growing environment, and hosting cloud-based agricultural management systems. For example, Fujitsu offers a system that measures the air temperature, soil temperature, moisture, solar radiation, and fertilizer levels in the soil using IoT sensors, transfers measurements to a cloud server at specified intervals and proposes optimal quantities of water and fertilizer for each farm based on data analysis and prediction. The farms using this system have experienced a small increase in yield. Because the production and yield can be predicted, the data collected by this system are also used to establish a procurement plan for agricultural products. A system for predicting vegetable yields is a representative system that is applicable to crops cultivated in open fields.

Table 2 and Table 3 present the main functions provided by representative

commercial environmental control systems that have been developed for smart farms to date in South Korea. Currently, environmental control systems based on the monitoring of crop growth have been developed as a domestic technology but have not been fully commercialized. On the other hand, other countries have already developed techniques for the active control of environmental conditions using crop growth monitoring systems. This technical gap is because growth and environmental data have been analyzed in depth for decades in leading countries, and these data have been successfully transferred to control systems using empirical formula-based modeling. AI technology is thus required for domestic Korean control systems in order to close the technical gap. This would enable the optimal use of the data collected from domestic crops, farms, and the environment.

Table 2. Comparison of smart farm control technology offered by a Korean company and an international company.

Functions of the climate controller	**Korean company**	**International company**
Environmental monitoring	○	○
Remote control	○	○
Crop monitoring	X	○
Type of control algorithm	P-band-based linear control algorithm	Nonlinear algorithm based on an environmental index (developed by Priva), development and application of a model based on an empirical formula
Precision of growth environmental control	Intermediate	Advanced
AI application	X	X
Energy efficiency	X	Partly

Table 3. Comparison of specific control functions between a domestic conpany and an international company.

	Details	Korean Company	International Company
Control logic	Temperature change management	○	○
	Six-phase control	○	○
	Windows step control	○	○
	Curtain gap control	X	○
	Temperature control for heat retention curtain with artificial light	X	○
	Curtain functional step	X	○
	Begin morning nautical twilight (BMNT) and end evening nautical twilight (EMNT) setting	○	○
	Delay setting	○	○
	Accumulated radiation consideration	○	○
	User settings for customization	X	X

The present study also conducted a literature review on the progress that has been made on control systems for greenhouses. The greenhouse environment can be divided into the atmospheric and the root zone environment. Based on this classification, relevant studies were reviewed, and the development trends for the prediction models and control systems were analyzed. The ultimate goal of the literature review was to consider the specific role of AI algorithms.

2.2. CONTROL MODELS FOR INSIDE CLIMATE IN GREENHOUSE

Kim et al., (2017) compares data-based models and physical energy equation-based models for greenhouse control systems. The authors also performed an experiment through P-Band control for field application. The data-based modeling turned out to be a possible alternative to the existing modeling method (Figure 11). However, since P-Band based control was necessary for real control operation, parameters had to be optimized, which seemed to be problematic. This problem could be solved by determining optimal Band coefficients in an iterative experiment.

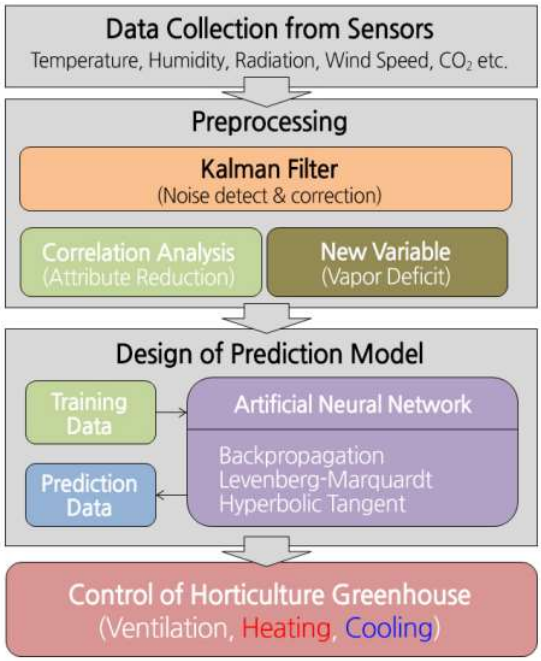

Figure 11. Frame of management system for greenhouse: Data collection, preprocessing, design of prediction model using ANN (proposed by Kim et al., (2017)).

Van Straten et al., (2000) argued that the environmental control of greenhouse requires an economic optimization strategy. The author also pointed out that the current P-band control had some limitations. According to them, it is necessary to develop models that can predict various environmental changes and be applied to various matters like crop modeling, photosynthesis and evapotranspiration. The development of a control and operation system, which could actively apply such

models, turned out to be important.

Coelho et al., (2005) introduced an optimal temperature control method for greenhouse under the concept of model-based prediction and control algorithm. The authors designed a prediction model by identifying outputs of a controller via the time series autoregressive method and proposed a method of determining control signals through a particle swarm optimization algorithm. They finally confirmed the better optimization performance of the particle swarm algorithm in comparison with genetic algorithms.

Kim (2018) considered agriculture in South Korea as an industrial sector where new values can be created by innovative AI technology based on machine learning like deep learning, IoT and big data. Accordng to Kim, as South Korea has entered into developing the second generation smart farm technolgoy, those who are not much experienced in agriculture will benefit from IoT, big data and AI technology, which would support decision making and automate environmental control. These technologies will advance South Korea's agriculture one step further.

Gouadria et al., (2017) developed a solution for greenhouse climate control employing an online PI tuned with PSO algorithm. They indicate that the greenhouse model has high strong coupling and non-linearity, a feedback-feedforward linearization and decoupling method employing measured disturbances was used. The present controller is made in terms of several performance criteria like settling time, peak overshoot, rise time. In addition to the responses due to step input. Results show that the tuning PSO-PI using in this work has a good performance.

Xu et al., (2018) defined a structural characteristic of Chinese type greenhouse system and proposed an optimal greenhouse energy control technique. The authors presented a two-time-scale receding horizon optimal control system that utilized the north wall structure of Chinese greenhouse to facilitate heat storage and discharge during the day and at night respectively (Figure 12). This model was developed by considering the climatic difference between the Netherlands and China. This case indicates the necessity of developing a greenhouse modeling and control system that is optimized to each country's environmental conditions.

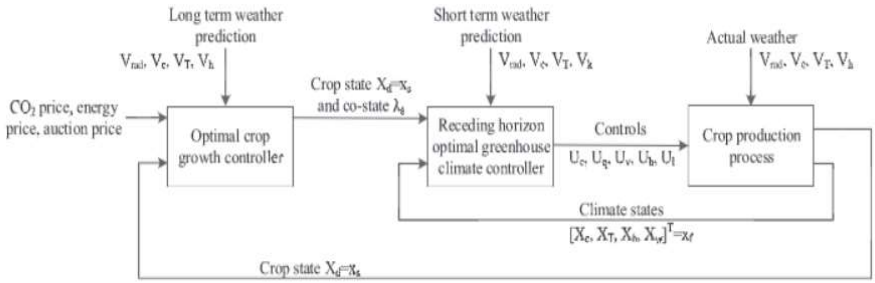

Figure 12. Two time-scale horizon optimal control system(TTRHOC) proposed by Xu et al., (2018)

Ding et al., (2018) survey on the development of MPC during the last forty years and describe the potential future application of MPC in agriculture. This study argues that the application of intelligent algorithms to modern agricultural production requires the support of a database, which can be complex and difficult to use in practice and requires a large amount of computing. Moreover, model predictive control (MPC) methods can achieve highly accurate control operations with moderate complexity and can also allow for rolling optimization in a limited time domain, which improves precision. MPC is highly suited for application in agriculture because it can effectively address nonlinear and large time-delay systems.

Muangprathub et al., (2019) proposed the wireless sensor networks' development for watering crops to optimize agriculture to design and develop the control system between node sensors in the field of crops and the data management via smartphone and web application. According to the results, the moisture content of the soil was maintained appropriately for vegetable growth, reducing costs and increasing agricultural productivity. Moreover, the data mining technique was applied to analyze the obtained data for predicting the suitable temperature, humidity, and soil moisture of crops in the future plan.

Yano et al., (2007) reported that a strategic ventilation strategy has an advantage in reducing energy consumption of greenhouse. Frequent control of vent openings according to the greenhouse temperature is difficult for common greenhouses where farmers operate vents by hand. Control of vent openings engenders advantages in greenhouse physical environment and energy consumption for the operation of vent openings is minimal.

Maher et al., (2016) presented for energy management by involving the photovoltaic energy in order to minimize the use of conventional electrical grid and to lower costs of agriculture production. The validation of the physical model showed a high agreement with the experimental measurement. The simulation results showed the effectiveness of the fuzzy controller as well as the PV generator for saving the energy and lowering the costs of crop production into greenhouses.

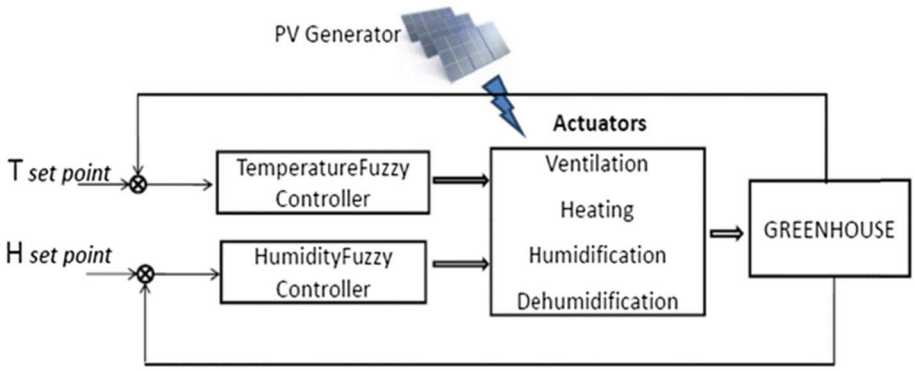

Figure 13. Schematic of the fuzzy-based control system (Maher et al., 2016).

2.3. DEEP LEARNING-BASED ENVIRONMENTAL MODELING

As a means of forecasting for a diversified society, the phrase ‘‘big data’’ has become widespread. Since such big data provides more insight than existing limited data, it has received greater attention in diverse research fields (Kim et al., 2017; Wolfert et al., 2017), including science, engineering, defense, management, medicine, and politics. Data modeling is a method in which a model represents correlation relationships between one set of data and the other set of data. On the other hand, physics-based simulation modeling is a more classical, but more powerful, method in which a model represents causal relationships between a set of controlled inputs and corresponding outputs (Kim et al., 2017).(Figure 14). However, the physical model has several limitations. For instance, there might exist situations in which we are able to know little or nothing about the system. Simulation requires extensive physical and operational knowledge of a target system in order to be accurate. In this condition, for prediction, we should adopt the approach of data

modeling explained in the previous section. In addition, the simulation model requires idealistic assumptions and constraints about the system, while the data model does not. Data-based modeling requires relatively short time for simulation and real-time field application, whereas simulation time for physical model is relatively long and rarely used in an actual application.

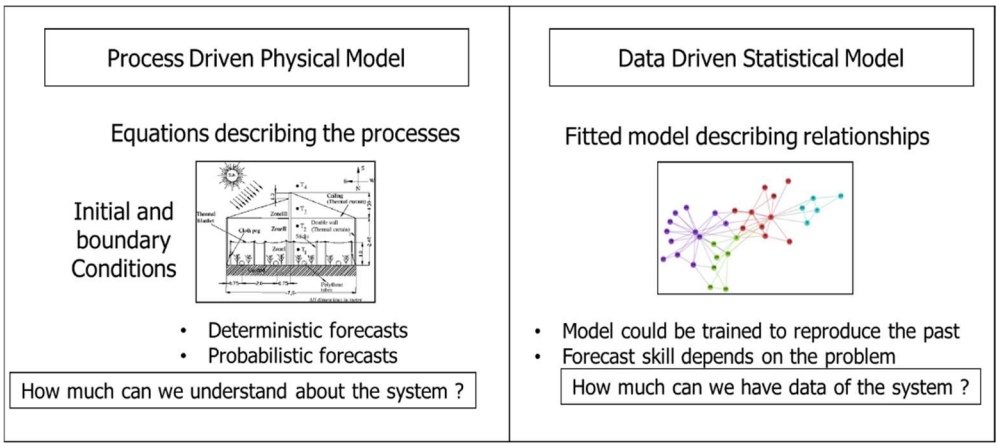

Figure 14. The difference between two modeling approaches.

Dodge, (2019) introduced a data science paradigm with the aim of advancing research on movement and compared the pros and cons of each application field of the data-based model in the existing physical modeling method.

Ferreira et al., (2002) used the radial basis function (RBF) NN to model the inside air temperature of a hydroponic greenhouse as a function of the outside air temperature, solar radiation, and the inside relative humidity. While the major drawback of such models, however, is their devoid of systemic analysis of input data samples of NN. In fact, large quantities of data samples often have redundant condition properties, data samples, and incompatible data samples. This cause many problems in the practical use of the NN models, such as slow convergence speed and low precision.

Taki et al., (2018) was aiming to select the best method between Artificial Neural Network (ANN) and Support Vector Machine (SVM) to estimate three different variables include inside air, soil and plant temperatures (T_a, T_s, T_p) and also energy exchange in a polyethylene greenhouse in Shahreza city, Isfahan province, Iran. According to study results, comparing RBF, MLP and SVM models showed

that the performance of RBF to predict three temperature variables were better according to small values of RMSE and MAPE and large value of R^2 indices. The range of RMSE and MAPE factors for RBF model to predict Ta, Tp and Ts were between 0.07 and 0.12 °C and 0.28−0.50%, respectively.

He and Ma, (2010) proposed A back propagation neural network (BPNN) based on principal component analysis (PCA) for modeling the internal greenhouse humidity in winter of North China. The environment factors influencing the inside humidity include outside air temperature and humidity, wind speed, solar radiation, inside air temperature, the open angle of top vent and side vent, and an open ration of sunshade curtain, which were all collected as data samples. The predicted humidity agreed well with the measured, which showed that the model had high accuracy and can be used. Moreover, they compared the methods of BPNN based on PCA and stepwise regression, it was observed that the BPNN based on PCA performed better than the stepwise regression model. This study emphasizes the importance of machine learning in future greenhouse environment models.

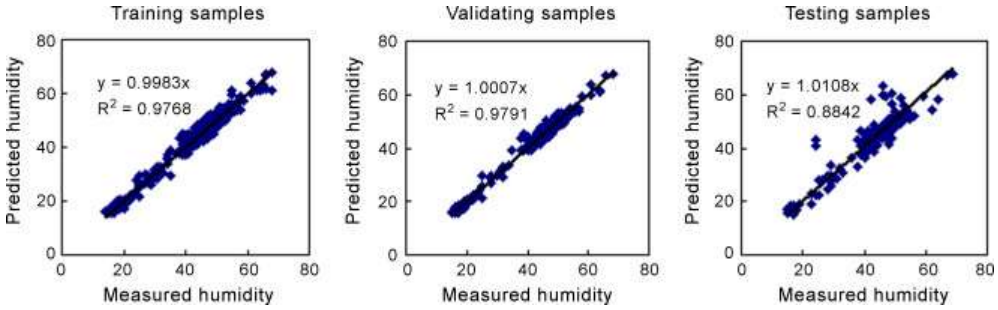

Figure 15. He and Ma, (2010) reorted result on regression lines between measured and predicted humidity by BPNN based on PCA.

S. L. Patil et al., (2008) applied neural network models in order to control greenhouse data that are continuously generated. Diverse models were comparatively verified, and the NNARX (neural network automatic regression) model had the best performance. However, the reliability of the model could not be ensured since it was reconstructed without verification of measurement errors and logging errors.

Unlike the modeling of atmospheric environment of greenhouse, that of the root zone has not been actively studied. Signore et al., (2016) emphasized the importance of a strategic approach to the control of nutrient solution replenished in order to

minimize the environmental problem caused by the hydroponic cultivation. In other words, a nutrient solution is not to be replenished in the same chemical composition, but its composition is to be changed according to the condition of the root zone substrate.

Pawlowski et al., (2016) proposed an Event-based control technique for analyzing greenhouse irrigation system. The event-based control scheme uses a crop transpiration model to determine the volume of water required to compensate for the irrigation system and a water content model to trigger the irrigation system events. They obtained results that the application of proposed event-based approach for the greenhouse irrigation system allows to improve the control performance and to reduce the water usage (about 20% of required water for the same performance obtained for commonly used on/off) being an important issue in intensive agriculture. However, this control method has a limitation that sophisticated evapotranspiration modeling should be preceded.

Whitley et al., (2009) incorporated the use of an artificial neural network (ANN) as a statistical benchmark to compare the performance of both modified a Jarvis-Stewarts model and the Penman-Monteith equation. Both models performed equally well during summer when soil water content was fairly high. During winter the modified Jarvis model performed significantly better than the Penman–Monteith model, especially under conditions of high transpiration. However, over the entire study period, the total modeled daytime sums of water used were all very close to the observed sum of 75.4 mm.

Yu et al., (2016) proposed an optimal prediction model that is based on LSSVM (least square support vector machine) for predicting temperatures of solar greenhouses in China. The authors also used PSO (particle swarm optimization) through an empirical model. An artificial neural network model was constructed to compare prediction performances. In addition, the data cleaning was also conducted by using the generalization method and averages in the preprocessing stage. However, the proposed prediction model was not applied to control a real greenhouse. In this regard, an appropriate greenhouse control strategy was required.

Wang et al., (2017) intended to obtain a transpiration model to implement greenhouse automatic water management based on the knowledge of the crop water

demand combined with automatic irrigation technology. In order to improve the prediction accuracy, they applied an optimization of the input vectors weights and accumulated error reduction. Although finally, the NARX has a better fit with actual transpiration, the result agrees with that of other research results (Sánchez et al., 2012). They emphasize that compared with transpiration model using NN, the precision could be improved.

Moon et al., (2018) applied the deep learning technology to predict the EC status of the root zone. It turned out that the RNN-LSTM model had the best prediction performance in comparison with various time series algorithms (Figure 16). This method innovatively improved the accuracy of environmental modeling for the root zone, as compared to the conventional complex modeling. Accordingly, it was also reported that diverse strategies for providing nutrient solution could be developed.

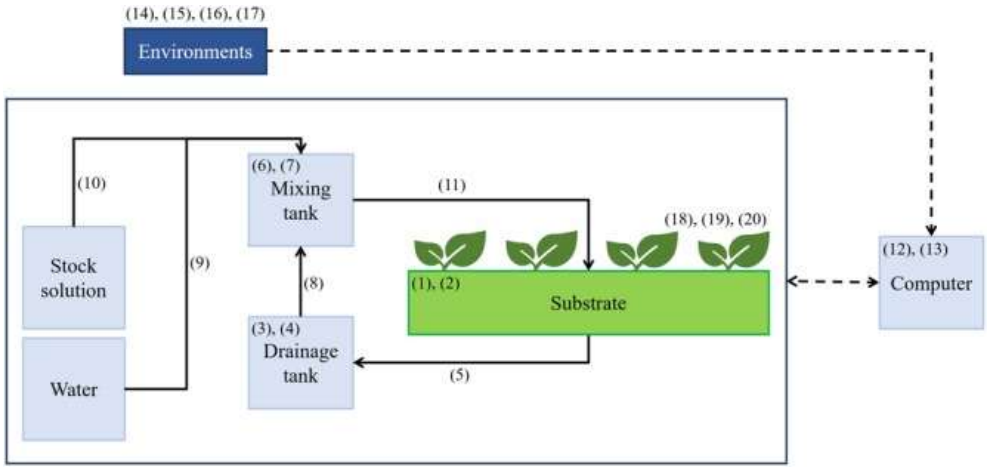

Figure 16. A diagram of a closed-loop soilless culture system and measured data of nutrient solutions and growth environment (reported by Moon et al., (2018))

Kim and Cho, (2019) proposed a novel deep learning model to stably predict electric energy consumption. The study analyzed the CNN-LSTM model with the large data collected in an actual residential house, and the model achieved the highest performance in high resolution compared with the previous works.

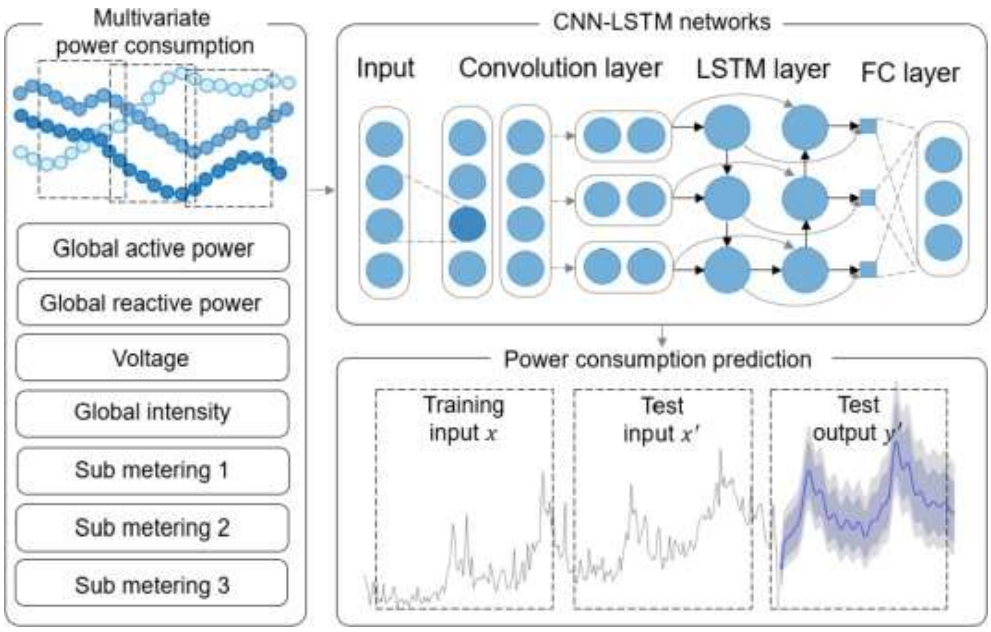

Figure 17. CNN-LSTM structure proposed by Kim and Cho(2019)

Livieris et al., (2020) proposed a new deep learning forecasting model for the accurate prediction of gold price and movement. The proposed model exploited the ability of convolutional layers for extracting useful knowledge and learning the internal representation of time-series data as well as the effectiveness of long short-term memory (LSTM) layers for identifying short-term and long-term dependencies. The input used in this model is also a one-dimensional array, and the model is implemented using 128 past data. As a result, CNN−LSTM reported 0.0082, 0.0095 and 0.01 MAE score for evaluating model's forecasting performances.

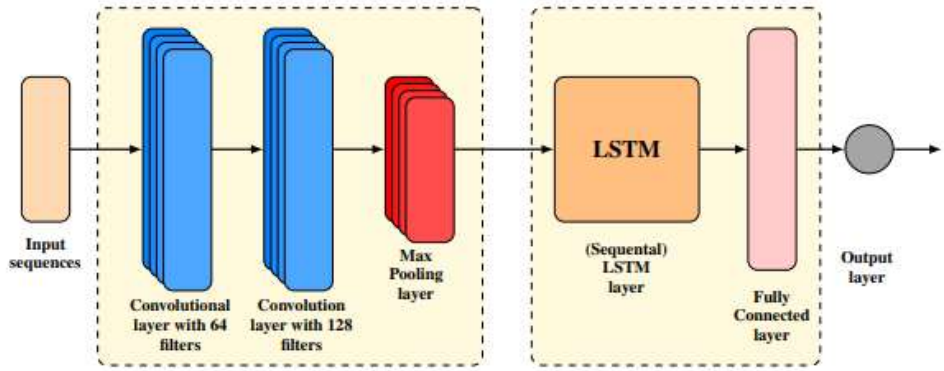

Figure 18 CNN−LSTM model architecture proposed by Livierise et al., (2020).

2.4. APPLICATIONS OF AI TO CLIMATE CONTROL

Fourati and Chtourou, (2007) attempted to solve the problems of the classic method in controlling and managing greenhouses. The authors trained an inverse dynamics model using a recurrent neural network and a multilayer-feed forward method and proposed a prediction model. The model thus constructed was verified through a simulation. They concluded that the artificial neural network was appropriate to control a complex greenhouse. However, an adaptive control method or a control strategy using a multiple neural network was required for enhancing control.

Kim et al., (2018) estimated heating loads through ANN-based prediction of an outdoor temperature of greenhouse. The reliability of data was ensured by applying preprocessing methods like IQR, Kalman filter and correlation analysis to measured data. The outdoor temperatures of greenhouse were predicted through an ANN model. The ANN model was verified by being compared with a multiple regression model and an SVM model with respect to the predictions. It turned out that the ANN model showed the best performance in spring and winter.

Calise et al., (2001) reported a direct adaptive output feedback control design procedure which is developed for highly uncertain nonlinear systems. This approach is also applicable to systems of unknown, but bounded dimension. In particular, the method considered single-input/single-output nonlinear systems, whose output has known, but otherwise arbitrary relative degree. This includes systems with both parameter uncertainty and unmodeled dynamics. The result is achieved by extending the universal function approximation property of linearly parameterized neural networks to model unknown system dynamics from input/output data. Numerical simulations of an output feedback controlled van der Pol oscillator, coupled with a linear oscillator, is used to illustrate the practical potential of the theoretical results (Figure 19).

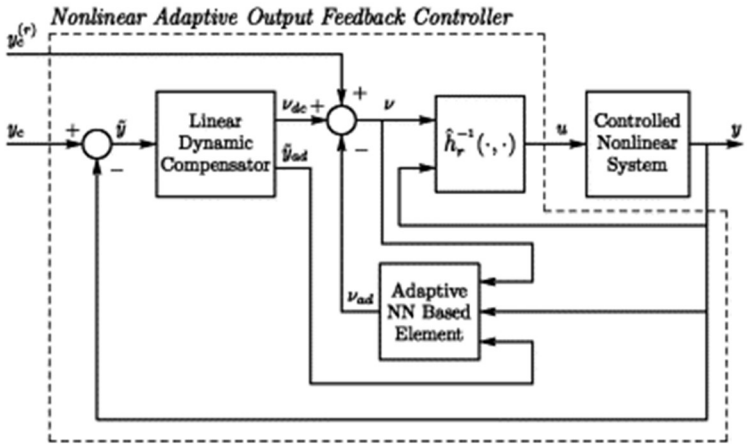

Figure 19. Nonlinear adaptive neural nettwork based control system architecture (proposed by Calise et al., (2001)).

Recently, Ban and Kim, (2017) suggested an application of actor-critic reinforcement learning approach to control a nonlinear, complex and black-boxed system. they demonstrated this approach on artificial greenhouse environment simulator all of whose control inputs have several side effects so grower cannot figure out how to control this system easily. this approach succeeded to maintain the circumstance at least 20 times longer than PID and Deep Q Learning. However, the model only performed in lab scaled greenhouse.

Tarange et al., (2015) developed web based automatic irrigation controller and reported Embedded Linux board. The board has an Ethernet interface and runs the simple data web server. Hence coordinator collects the data over ZigBee wireless communication protocol and allow user to monitor the data from a web browser.

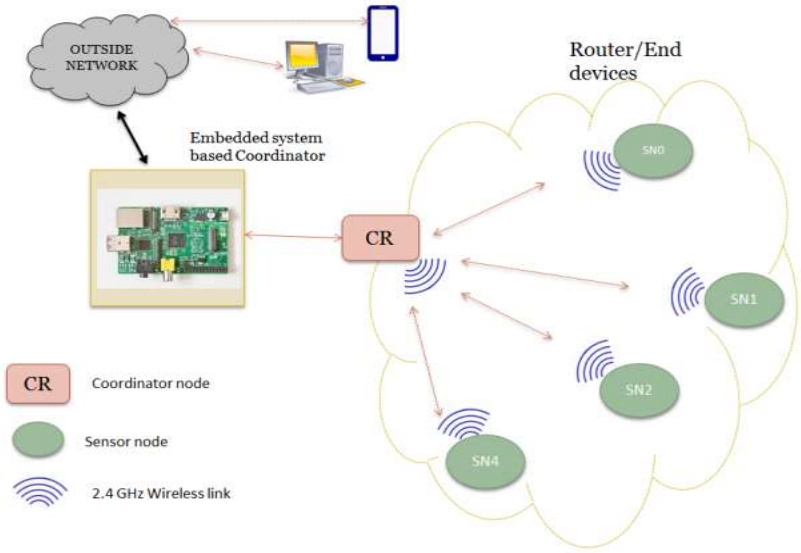

Figure 20. Automatic irrigation controller and reported Embedded Linux board board (developed by Tarange et al., (2015))

3. DEVELOPMENT OF MULTIVARIABLE CLIMATE CONTROL SYSTEM BASED ON LINEAR ALGORITHM

A greenhouse is a common agricultural system that artificially manipulates the growth environment for a target crop. Maintaining an optimal temperature, humidity, and CO_2 levels are of particular importance in greenhouse environmental control because these environmental factors directly affect plant development, quality, and quantity. The greenhouse climate system is very complex and nonlinear (El Ghoumari et al., 2005; Seginer and McClendon, 1992), with the variables highly dependent on external climate conditions, which cannot be controlled independently (Fourati and Chtourou, 2007a; Frausto et al., 2003), and on greenhouse design. Crops exposed to low or high temperatures or humidity may also undergo mass death or drying due to fungal diseases and the lack of water, leading to significant financial losses for growers. Therefore, building a precise model of the internal greenhouse climate is important when looking to respond to these dynamic changes and to establish an efficient climate management strategy (Fitz-Rodríguez et al., 2012; Yu et al., 2016a).

Lanfang et al. (2000) described two approaches to building models of the greenhouse climate. One is based on the physical laws involved in the process, and the other is based on an analysis of input–output data for the process. In recent decades, due to an increase in computational performance, numerous physical greenhouse methods have been presented (Benni et al., 2016a; Kishor and Singh, 2007; Norton et al., 2007; Soldatos et al., 2005), but this approach may yield inconsistent results when applied to real-world conditions. This is because the models are defined by high complexity and often need to calculate and estimate unmeasurable parameters such as the soil heat flux density, biological factors, photosynthesis rate, water vapor pressure, and environment changes. Prediction methods based on data processors have been proposed and applied in various fields due to the development of modern computational technology. This approach provides rapid and accurate results for implementation in the agricultural industry, such as the prediction of the internal temperature inside a greenhouse (Nury et al.,

2017; Patil et al., 2008), transpiration rate prediction (Wang et al., 2015), and ventilation strategy in livestock (Soldatos et al., 2005).

ANN models represent a powerful forecasting tool for analyzing nonlinear systems because of their ability to model systems without the need to make any assumptions, which are often implicit in most traditional statistical approaches (Dariouchy et al., 2009). For this reason, these have been applied to the prediction of greenhouse climatic data and generally produce better results than state-of-the-art physical models (Fitz-Rodríguez et al., 2012; He and Ma, 2010; Nury et al., 2017). With this method, however, some concerns have been raised about optimization and application in practical use, including over-fitting, the need for many training sets, the lack of flexibility between different crops, and occasionally poor stability in strongly coupled and complex systems.

The temperature, humidity, and CO_2 levels inside a greenhouse are strongly coupled to system elements, such as ventilation, heating, fogging, and other actuators and physical behaviors in the greenhouse. Another obstacle to the development of precise models is the fact that greenhouses have different actuators for each target climate variable, and these actuators typically exhibit simple on/off behavior, thus a single element can be a dominant factor that leads to changes to the internal climate of the greenhouse. Past research has proposed time-series models that are able to provide simplified representations of large numerical systems for the accurate simulation and prediction of their dynamic responses (El Ghoumari et al., 2005; Fan et al., 2012; Lu et al., 2014).

Greenhouse operating systems have proposed index criteria for heating and ventilation loads, and these systems employ a variety of elaborate experience-based models. These models make appropriate decisions based on crop growth data and greenhouse environmental information that have been accumulated over the past several decades (Yang et al., 1989). However, in countries with different climate conditions, the optimal value of the coefficients used in proportional band (P-band) control must be obtained using trial and error, or farmers can operate the greenhouse control system using their own established knowledge. Domestically, commercial products that utilize logic based on P-band control have been developed. P-band control uses a simple linear model that is applied to influence coefficients, which are

related to other influential factors that can affect the target environmental factor. For example, solar radiation, the outdoor temperature, and wind speed are commonly monitored as influential factors for ventilation control. However, it is very difficult to determine intuitively what the exact values should be for the coefficients for these factors and whether they have a positive or negative influence.

To identify these conditions, experimental statistical methods have been widely used (Majdi et al., 2019). Of these methods, response surface methodology (RSM) determines optimal coefficients based on various experimental conditions (Thakur et al., 2018). This method minimizes the number of scenarios required for the experiment, so it is widely used in many fields (Kaushal et al., 2015). In addition, RSM is frequently used for tuning coefficients in the proportional–integral–derivative (PID) controller for many applications (Demirtas and Karaoglan, 2012; Nakano and Jutan, 1994). Research on the optimization of coefficients for greenhouse environmental control using an experimental statistical approach is required, but actual results for this have rarely been reported.

As an example, in greenhouses that utilize natural sunlight, optimal influence coefficients must be determined according to time. Daily changes in the greenhouse environment tend to follow certain patterns. The indoor temperature begins to rise from sunrise, and the heat inside the greenhouse tends to accumulate until the afternoon. However, the temperature inside the greenhouse before sunset drops drastically, meaning the heat is not preserved overnight (Blasco et al., 2007; Kwon et al., 2013). Therefore, a day can be divided into sunrise, sunset, noon, and midnight periods, and different strategies must be devised for these time periods in order to precisely control the greenhouse environment. A commercial integrated climate control software system for greenhouses should thus allow the user to change the settings for each time period, and there is a need to identify the corresponding optimal influence coefficients

The specific objectives for this chapter are to

1) Explain the components, structure, and system of the target smart farm, and explain the specific data collection method and installation process required for the research.

2) Introduce linear control (proportional derivate band [PD-band]/P-band) as the logic for greenhouse climate control, and
3) Conduct an optimization experiment for outside influences on linear based P-band control.

3.1. DESCRIPTION OF CLIMATE CONTROL SYSTEM IN SMART GREENHOUSE

An experimental greenhouse at KIST, Gangneung, constructed as a multi-span Venlo structure, was used for this experiment. The size of each unit was 16 m in length and 16 m in width, and a total of four units were connected together as shown in Figure 21. Excluding the working space, the actual planting area was 16 m in length and 12.5 m in width, covering an area of 192 m^2. The greenhouse was located at 37°47'47.1" N, 128°51'24.5" E.

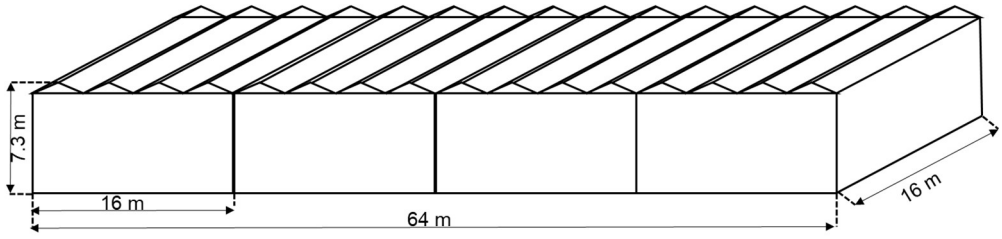

Figure 21. Overall appearance of the multi-span venlo-type greenhouse.

3.1.1. HARDWARE OF CONTROLLER

The crop used in the experiment was Dafnis tomatoes. A sensor node was located at the center of the greenhouse (Figure 22a) and the data collected every ten seconds was averaged each minute and then saved to a database. The sensor node measured the temperature, humidity, and CO_2 levels inside the greenhouse. A DS Thermistor (R-T type, ADS103FU-C) was used to measure the temperature and humidity, and an SH-300-DC (Soha Tech, Korea) was used to measure the CO_2 concentration. The experimental greenhouse was operated using an algorithm developed in-house.

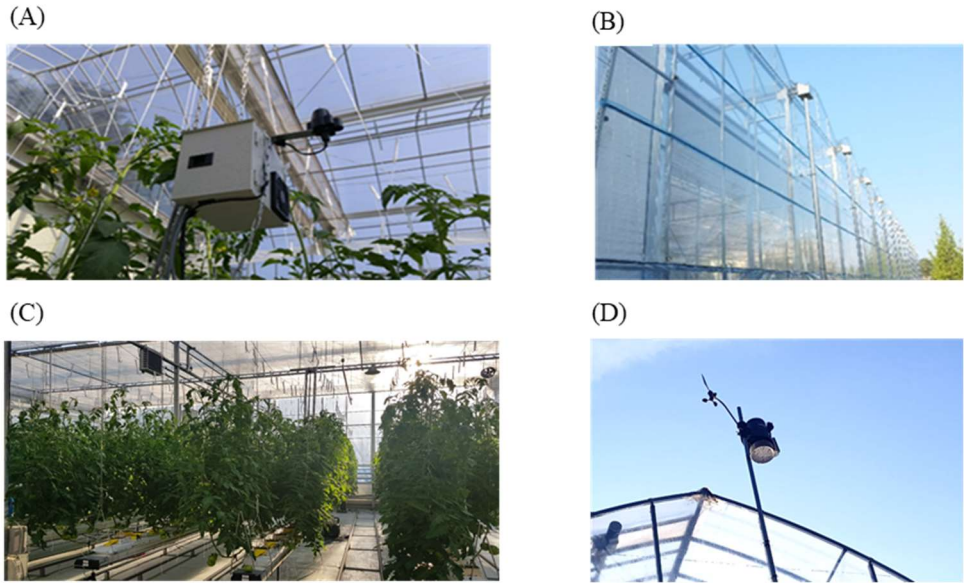

Figure 22. Experimental set-up. (a) Temperature sensor used to monitor the environment inside the greenhouse. (b) Outside view of the vinyl multi-span Venlo greenhouse. (c) Tomatoes growing inside the greenhouse. (d) Weather station to measure outside climatic variables.

A climate control system was installed to manage the environment inside the smart greenhouse (Figure 23). Each actuator was driven by a separate control logic and setup, operating according to the temperature, humidity, and CO_2 concentration inside the greenhouse.

The external and internal environmental data were collected by the weather station and the sensor node inside the greenhouse, respectively. The collected data was sent to the controller every minute using a standard transmission protocol. The transmitted data was stored in the database, which included a control algorithm program that generated operation commands for the actuators based on these data. The generated commands were then sent to a microcontroller, and the controller sent the signal to the relays to operate the actuators.

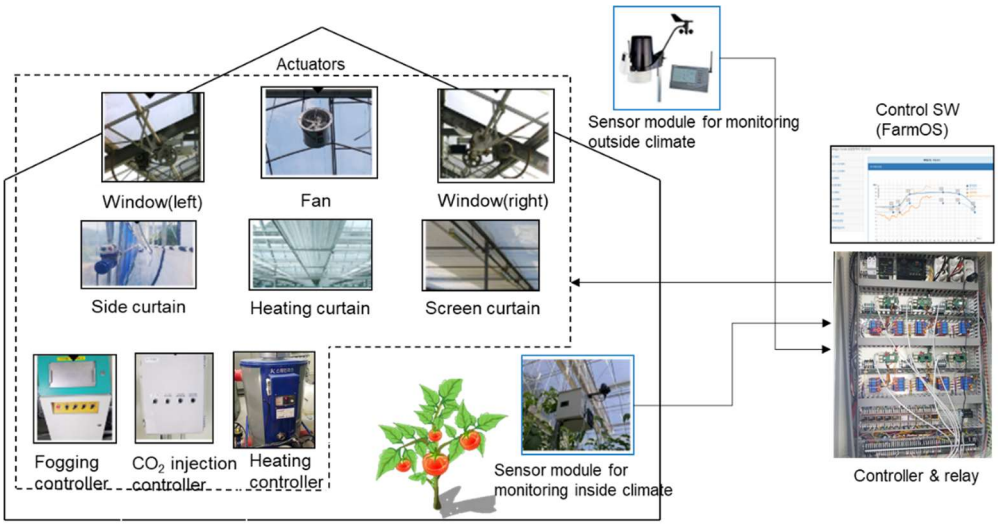

Figure 23. Monitoring sensors and environmental control actuators included in the greenhouse used for data acquisition.

A Raspberry Pi computer (Model B, Raspberry Pi Foundation, United Kingdom), which is cheaper and more accessible than other commercial products, was used as the main controller. Details and specifications for Raspberry Pi are provided in Figure 24 and Table 4. Data was transmitted from the sensor modules inside and outside the greenhouse and stored on the Raspberry Pi board, before being communicated to the cloud server and the installed relay board as shown in Figure 25.

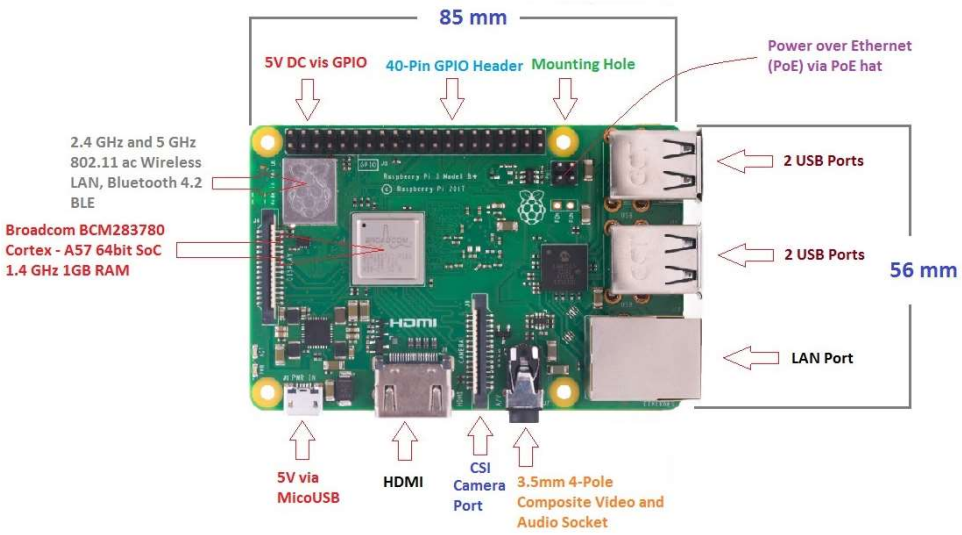

Figure 24. Features of the Raspberry Pi 3.0 model B.

Table 4. The specification of Raspberry Pi 3.0 model B

	Raspberry Pi Model B
Release date	Feb. 29, 2016
SoC	BCM2837
CPU	Quad Cortex A53 @ 1.2GHz
Introduction set	ARMv8-A
GPU	300MHz videocore IV
Ram	1 GB SDRAM
Storage	Micro-SD
GPIO	40
Wireless	802.11n/Bluetooth 4.0
Price	$35

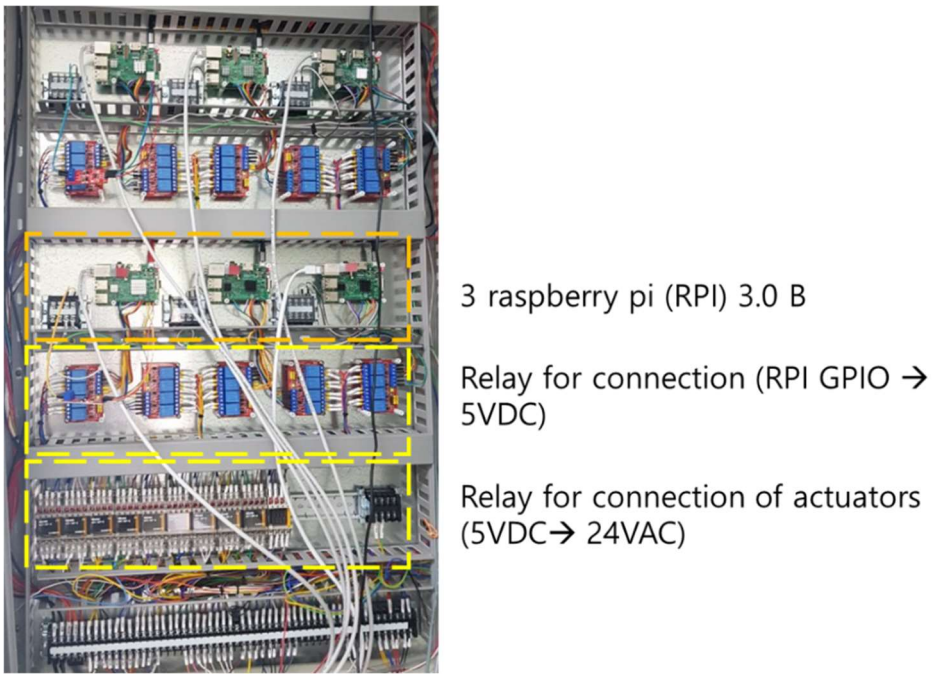

Figure 25. View of controller and relay board installed in the control room of the greenhouse.

The floor plan for the greenhouse, including the position of the actuators and all relevant equipment (e.g., the fogging injector), is shown in Figure 26. The four individual greenhouse units were continuously connected. Figure 26(a) shows the location of the controller, the program that drove the controller, and the computer on which the program was installed. Figure 26(b) indicates the control box that connected the power supply to the greenhouse equipment (e.g., windows and curtains) and that connected the 24VAC power from the control room controller to operate the relay. Details of the operation and connection of the actuators are described in detail in Section 3.2.

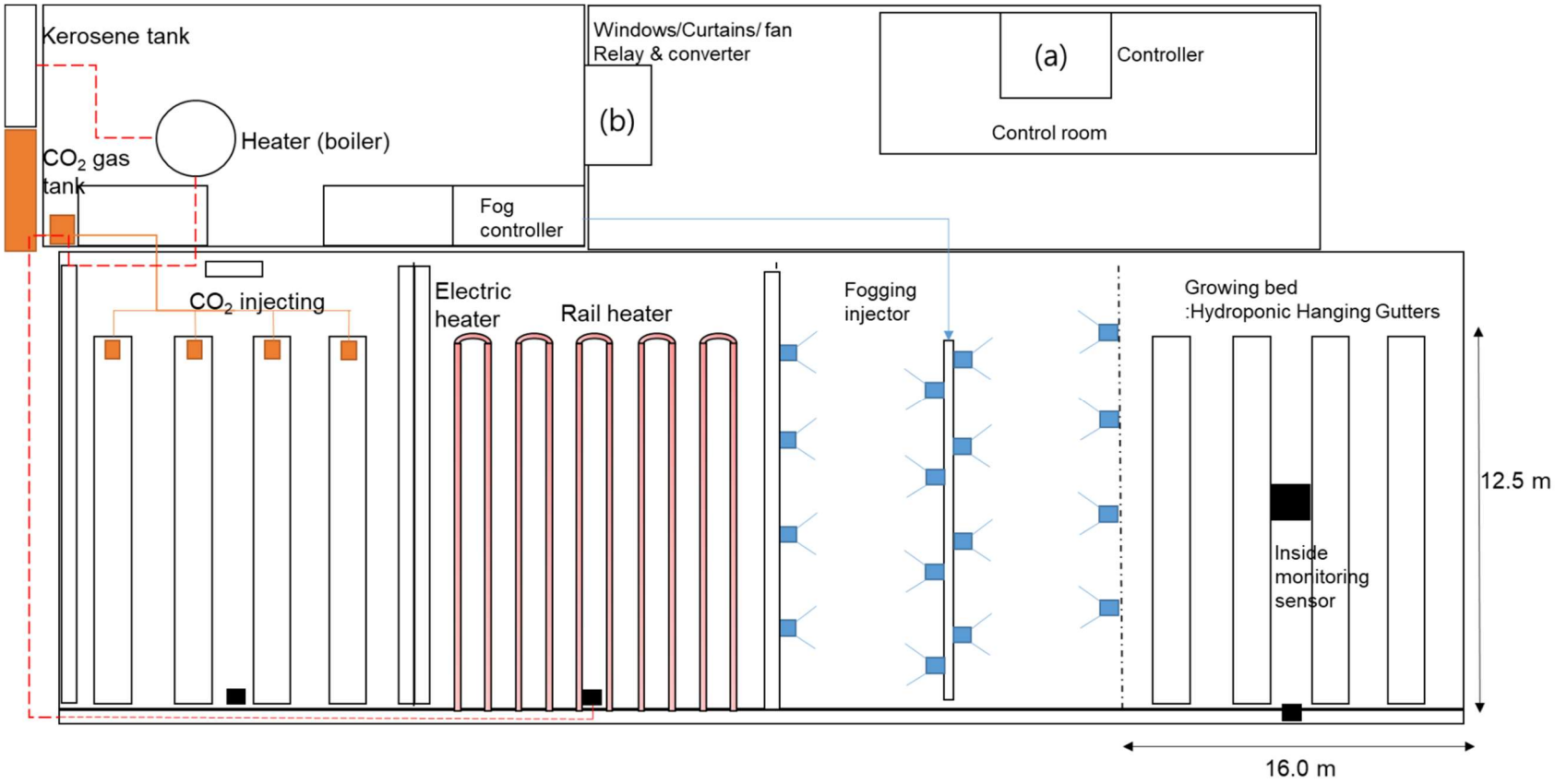

Figure 26. Floor plan showing the structure of the greenhouse and the position of the controller and actuators

3.1.2. CONTROL SOFTWARE

Two control programs were employed in this study. First, the self-developed control program Welgrow was used to implement the control logic. It was eventually replaced with operating software developed for the purpose of monitoring and controlling the integrated environment of a greenhouse based on the open platform FarmOS (FarmOS V2, Jinong Inc., Gyeonggi Province, Republic of Korea) (Figure 27). The FarmOS-based software was then migrated and the actuators and sensors were modified to match the standard communication protocol (Huh et al., 2018). The Raspberry Pi computers were configured to meet KS X3267 communication standards (Park et al., 2019) by acting as sensor nodes and control nodes (Figure 28).

A sensor module (SH-VT250, Soha Tech, Korea) was used to monitor the internal climate of the greenhouse. The sensors were installed at the center of the greenhouse, and the specifications for the sensors are presented in Table 5. The environmental controller consisted of a sensor node and a control node for the processing of sensor data. The software program that operated the individual nodes was installed on a Raspberry Pi computer (Model B, Raspberry Pi Foundation, United Kingdom). Monitoring and control logic were implemented using an open platform program (Park et al., 2019). The control logic used a control algorithm based on a P-band. The P-band used for ventilation determined the opening of the windows (%) by calculating the ventilation load using the external temperature, wind direction, wind speed, and solar radiation as setting parameters and linear coefficients. All of the control signals and sensor data from the environmental controller were stored in the database and used for the development of the prediction model and the design of the control algorithm.

Table 5. Sensor specifications for the internal climate of the greenhouse.

Component	Measurement range	Resolution	Operating temperature (℃)	Response time (s)
Temperature	-10.0–50.0 ℃	± 0.3℃	-25.0–85.0	5.0–30.0
Humidity	0–99.0 % RH	± 2.0 %	-10.0–50.0	8.0
CO_2	0–3000 ppm	± 10.0–50.0 ppm (proportional to the measurement range)	-10.0–0.0	2.0

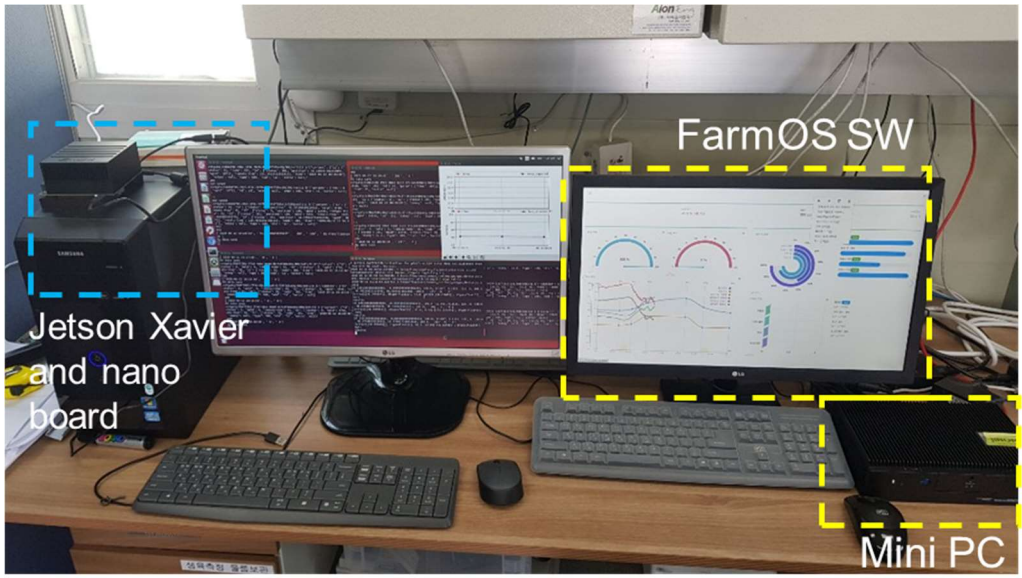

Figure 27. Environmental control program monitoring in greenhouse operating room.

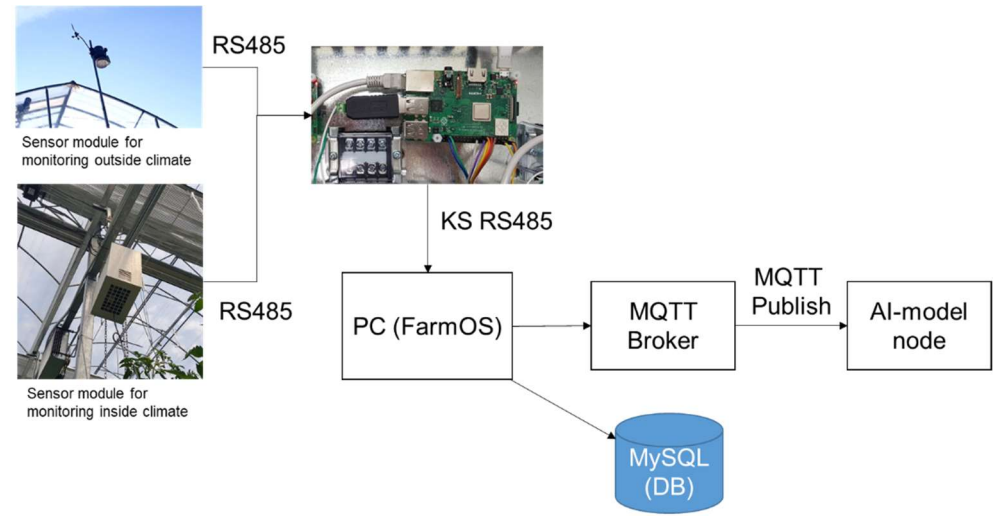

Figure 28. Overview of monitoring sensor communication protocol

A message queuing telemetry transport (MQTT)-based message transmission protocol was used so that the collected environmental information could be received by authorized devices. MQTT is a lightweight open OASIS and ISO standard (ISO/IEC 20922) publish–subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP; however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. The MQTT protocol is based on the principle of publishing messages and subscribing to topics of interest.

The sensor data installed inside the greenhouse was collected through the Raspberry Pi sensor node, reprocessed, and published to the MQTT broker. The MQTT broker can be accessed by multiple clients who subscribe to the information. In the present study, the online automatic learning and environment prediction system was configured to be connected as a single client, as shown in Figure 29. The client was a small microcontroller (Jetson Nano, Nvidia, USA) and installed based on an embedded board with an Ubuntu 18.04 OS. The configured MQTT broker published environmental information from the greenhouse, the driving signal from the controller, and the current location every minute to deliver the information, and the embedded board composed of clients was given a command to collect this information and store it separately.

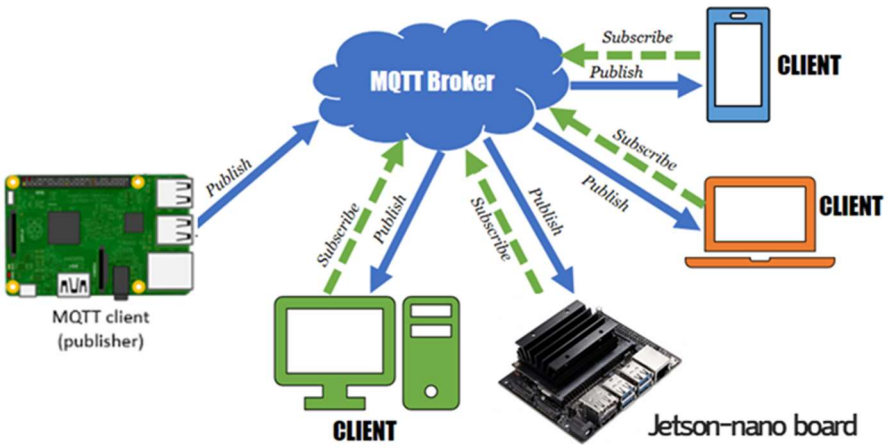

Figure 29. The overview of MQTT data publish/ subscribe system between the broker and clients

The control program was an integrated greenhouse climate control algorithm that was developed in-house for the present study. Rule-based operations could be conducted for both P-band- and PD-band-based algorithms using this program. Table 6 presents the rules for various environmental conditions, while Figure 30 displays the environmental control program for the Welgrow version and the setting window information for each section, and Figure 31 shows the FarmOS-based dashboard.

Table 6. Control operations for the environmental control program used in this study

	Ventilation control	**Heating control**	**Curtain control**	**CO_2 control**	**Fogging control**	**Flow fan**
Operating conditions	Difference between the set temperature and the indoor temperature	Difference between the set temperature and the indoor temperature	Solar radiation/ outside temperature	Indoor CO_2 concentration	Indoor temperature condition	Ventilation status
Actuator logic	PD band	PD band	P band	P band	P band /Trigger	Trigger
Influencing factors	Solar radiation, wind direction, wind speed, outside temperature	Solar radiation, outside temperature	Solar radiation, outside temperature	Outdoor CO_2 concentration and solar radiation when ventilating		
Constraint controls	Close all sections of the roof when raining Open all windows slightly (5%) during heavy winds Close all windows when spraying pesticide					

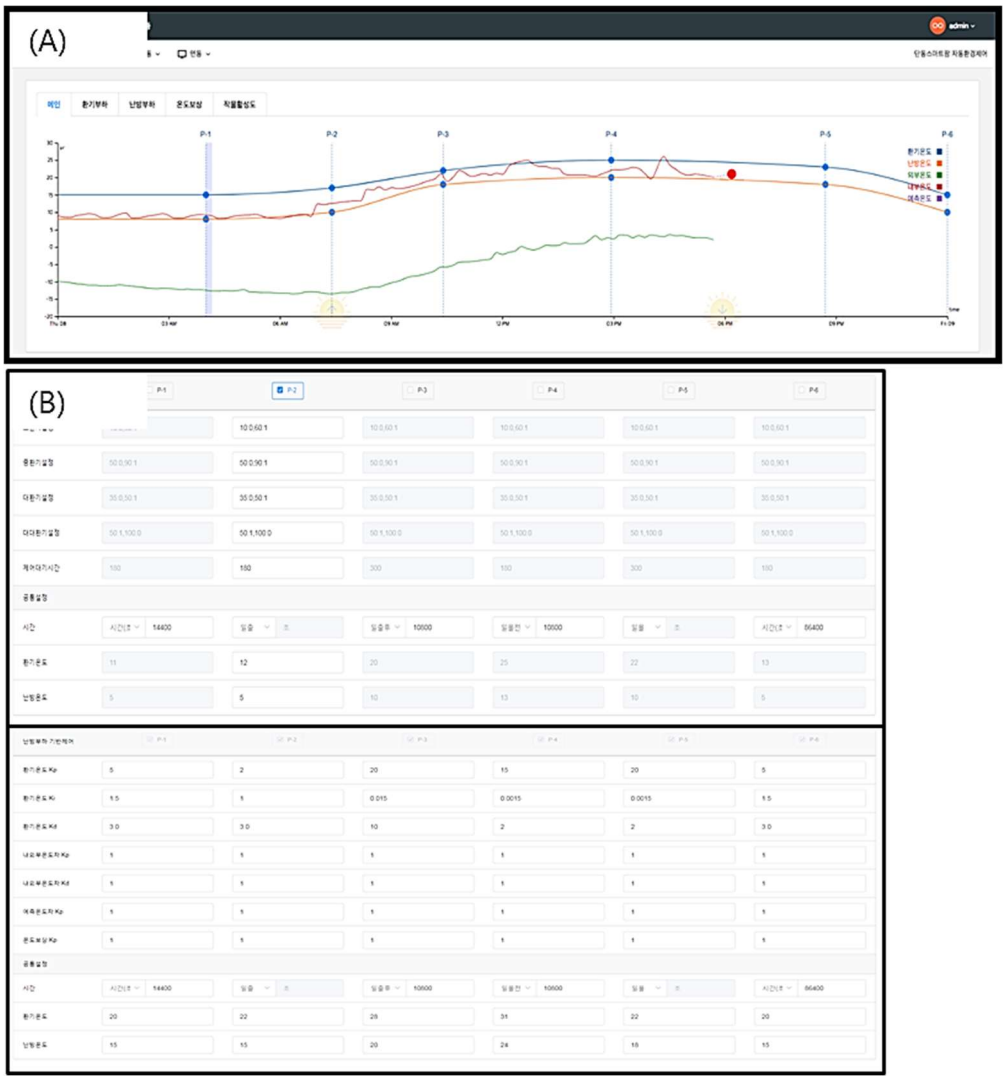

Figure 30. Climate control software (Welgrow) developed for smart farm control in this study: (A) monitoring screen and (B) setting screen.

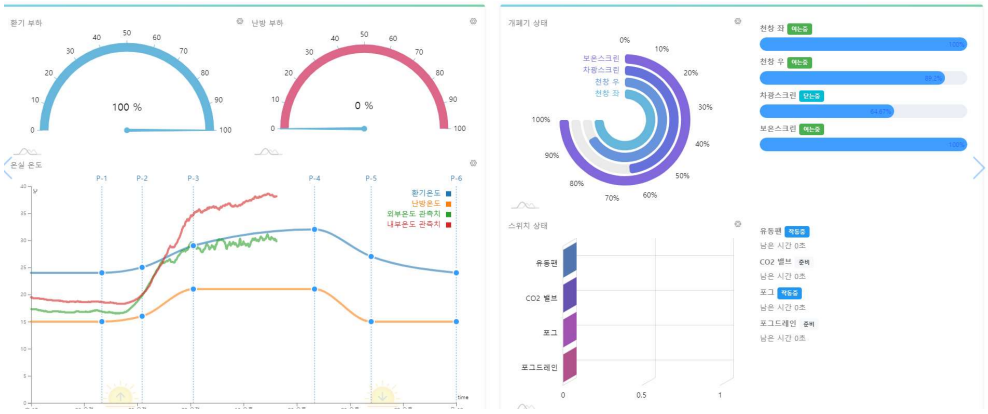

Figure 31. creen view of FarmOS, which can be used as an open platform for greenhouse climate management.

3.2. MULTIVARIABLE CLIMATE CONTROL BASED ON A LINEAR ALGORITHM

3.2.1. PD-BAND BASED CONTROL LOGIC FOR THE VENTILATION CONTROL

In this study, the PD-band control technique was introduced as the logic for greenhouse ventilation control. In order to optimize the settings for this control logic, research was conducted to obtain the values for the solar radiation, outside temperature, and wind speed, as well as the values for the optimal influence coefficients for the P-band and D parameters. Thirty-two experimental conditions were derived from the RSM analysis, and a day was divided into six time periods so that the coefficients could be optimized for each time period. Ultimately, this study proposes a very simple and efficient method for temperature control through ventilation in the greenhouse and discusses the optimal coefficients for each time period.

Ventilation control allowed external air to flow through the skylight in the experiment greenhouse. The structure of the window is shown in Figure 32 . Four windows were installed in each greenhouse unit, two on the left and two on the right, and windows in the same direction were opened or closed simultaneously. To open

and close the window, the installed rack gear was operated with a gear motor (CODM81064, Chung-oh Engineering, Daejeon City, Korea). The specifications for the motor are as presented in Table 7.

Table 7. Specifications for the gear motor installed to open and close the windows in the greenhouse.

Description	Value
Model number	CODM 81064
Voltage	DC 24 V
Rated current	8.0 A
Rated power	192 W
Motor speed	1.5 RPM
Reduction ratio	1:2240
Opening height/distance	4/100 m
Weight	7.6 kg

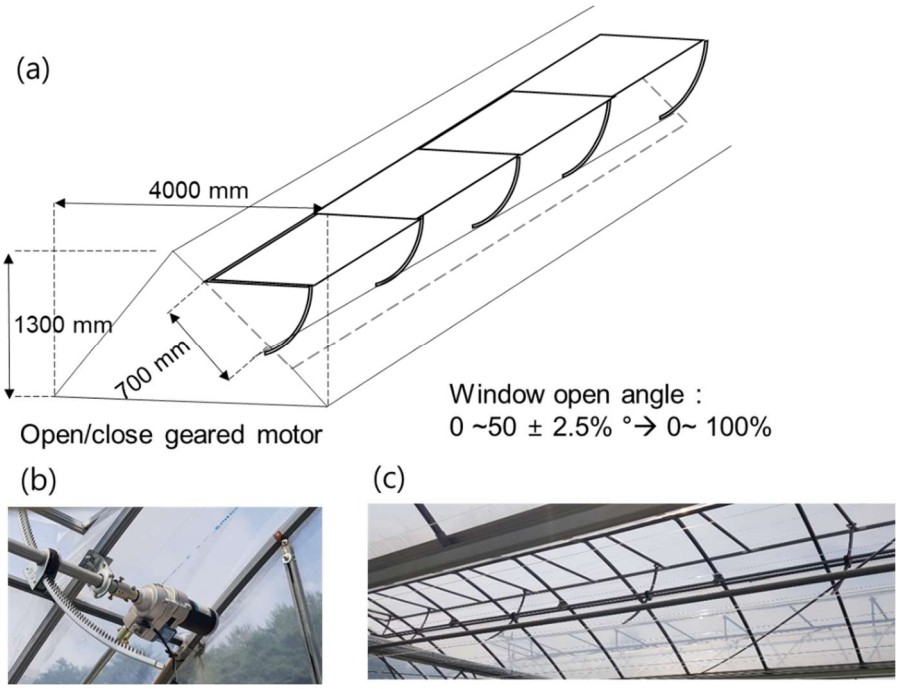

Figure 32 (a) Opening of the windows for ventilation control. (b) Installation of the rack geared motor to open the window. (c) Overall view of the windows.

Each window could open to about 50 degrees, which was defined as 100% open (Figure 32b and c). The opening and closing (100% and 0%, respectively) times were recorded in advance to adjust the position based on time. At the maximum open and closed positions, a limiter was set to stop the motor. A P-band is a control logic that is widely used in integrated greenhouse climate control systems. In this logic, the controller can decide how far to open the ventilation windows within the range of 0% to 100%. For example, when the greenhouse temperature is measured, the ventilation windows are opened to a degree proportional to how much the actual temperature has exceeded the set temperature by. Here, P-band control is required to calculate the position of the open ventilation windows.

The P-band indicates the degree to which the actual temperature needs to exceed the set temperature for the windows to be opened 100%, expressed in Celsius. Thus, once the greenhouse temperature exceeds the set temperature, the windows are opened by a certain percentage each time the greenhouse temperature rises by 1℃. Equation [1] was used to calculate the band value. In addition, the speed at which the heat and humidity were reduced using ventilation was determined primarily by the outside temperature and the wind velocity. The outside temperature is low and the wind is strong during winter, so the heat and humidity can be lowered quickly. However, during spring or early summer, when the outdoor temperature is high and the wind is relatively weak, it takes longer to reduce the heat or humidity. In order to seamlessly control the ventilation, the excess heat or humidity must be removed at an optimal speed (Fitz-Rodríguez et al., 2010). If greenhouse ventilation is too cautious, it takes longer to reduce the heat or humidity, which will rise as a result. In this case, a relatively small P-band is advantageous. However, if the P-band is set too low during the night when the heat needs to be preserved, the heat or humidity may be reduced too quickly, reducing ventilation performance. To avoid this issue, time differentials for the set value and the target value were added as constants to respond to the changes in the temperature difference.

$$\mathrm{PB} = X_1 + X_2 \cdot \frac{R_m}{R_c} + X_3 \cdot \frac{T_{om}}{T_{oc}} + X_4 \cdot \frac{V_m}{V_c} \qquad \cdots [1]$$

$$\text{Output } (\%) = \frac{100}{PB} \cdot (E + X_5 \cdot \frac{dE}{dt}) \quad \cdots [2]$$

X_1–X_5: Influence coefficients

PB: Proportional bandwidth index

R_m: Maximum solar radiation (set value)

R_c: Current solar radiation (measured value)

T_{om}: Maximum outside temperature (set value)

T_{oc}: Current outside temperature (measured value)

V_m: Maximum wind speed (set value)

V_c: Current wind speed (measured value)

E: Difference between the current temperature and the target temperature

T_c: Current indoor temperature.

In this study, the actuator controls can be classified into two categories as shown in Figure 33: proportional control actuators that can adjust the signal proportionally and actuators that can only turn on or off.

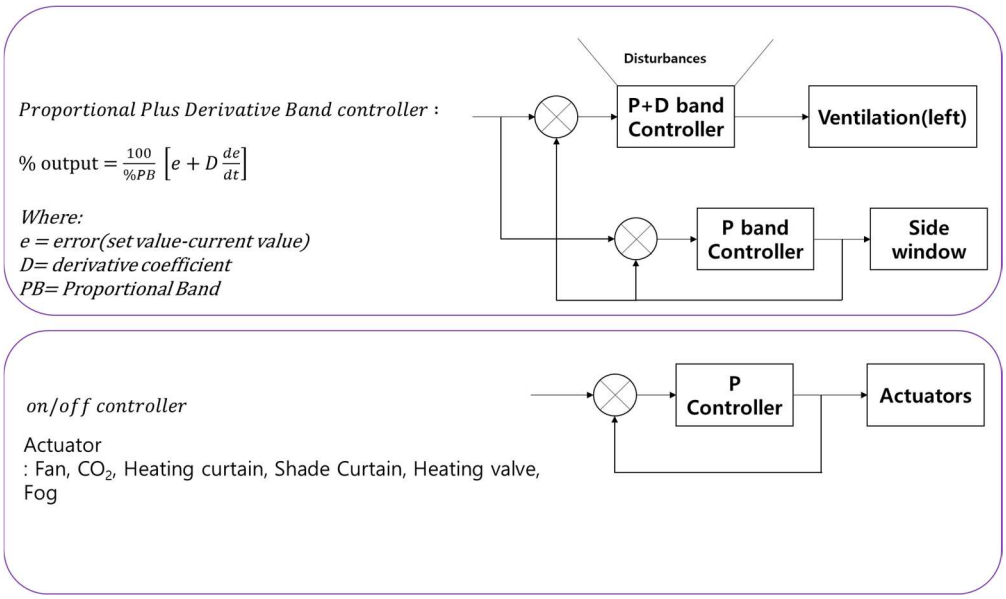

Figure 33. The two types of greenhouse control actuator employed in the present study.

3.2.2. CO_2 CONTROL

Maintaining CO_2 levels is essential to ensuring optimal plant growth; in the greenhouse in the present study, these levels were based on those at noon, when the photosynthesis of the crop was most active. A solenoid on/off valve was installed in the greenhouse connected to CO_2 cylinders; CO_2 was injected into the greenhouse through an inlet at the bottom of the growing bed (Figure 34). Equations [3] and [4] were used to calculate the band value for CO_2 control. The target CO_2 concentration was set at 500 ppm (user input) from 9:00 to 15:00, the period of highest photosynthesis activation. However, the actual control performance was weaker than expected, with a root mean square error (RMSE) of 35.44 ppm.

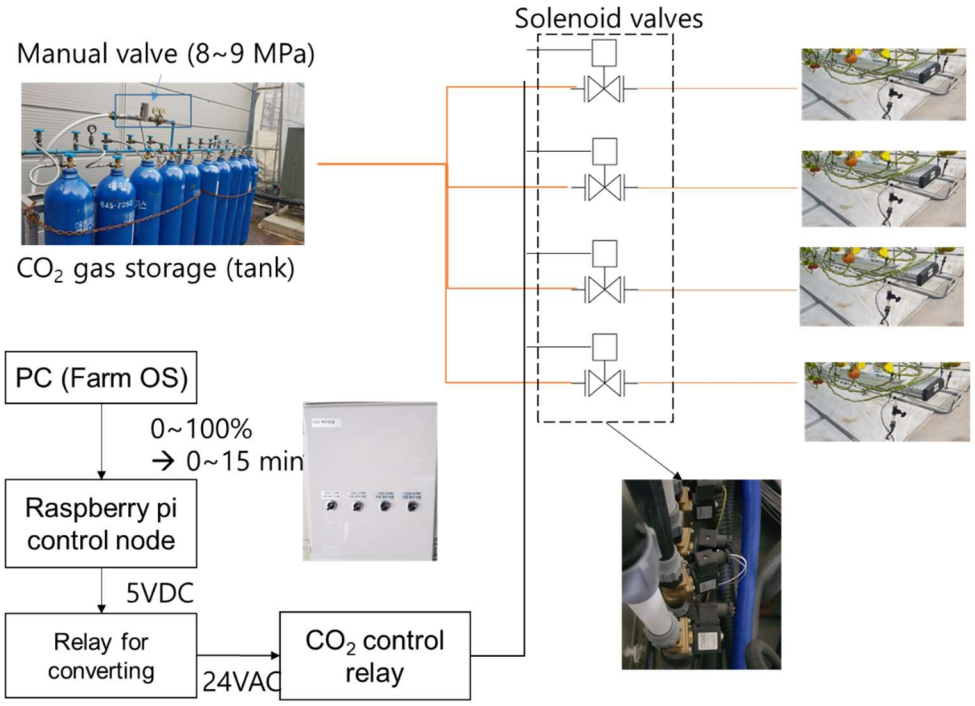

Figure 34.Carbon dioxide gas for spraying in the greenhouse and carbon dioxide inlet at the bottom of the growing bed.

$$\mathrm{PB} = X_1 + X_2 \cdot \frac{C_{om}}{HC_{oc}} \quad \cdots [3]$$

$$\mathrm{Output}\ (\%) = \frac{100}{PB} \cdot (E) \quad \cdots [4]$$

X_1–X_2: Influence coefficients

PB: Proportional bandwidth index

C_{om}: Maximum outside CO_2 concentration (set value 490 ppm)

C_{oc}: Current CO_2 concentrations (measured value)

E: Difference between the current CO_2 concentration and the target CO_2 concentration.

Table 8. Influence coefficients for the designated time slots for CO_2 control.

	X_1 (P base)	X_2 (Outside CO_2)
P_1	0	0
P_2	0	0
P_3	4.0	-1.0
P_4	4.0	-1.0
P_5	3.0	-10
P_6	0	0

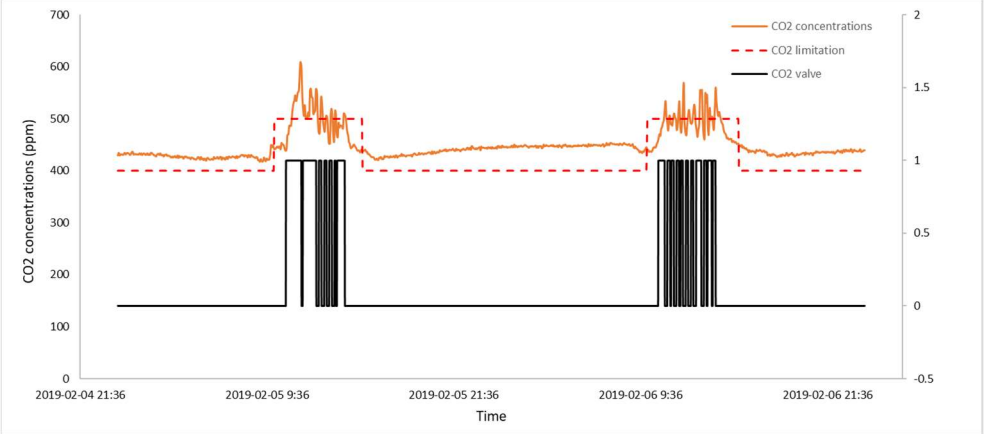

Figure 35. The results for CO_2 concentration control in the greenhouse.

3.2.3. HUMIDITY CONTROL

The relative humidity inside a greenhouse influences crop stress. When the humidity is below a certain level, the pores of the leaves close to prevent photosynthesis. In contrast, if it becomes too humid, mold can form on or around the crop, making it vulnerable to disease. The humidity levels in a greenhouse can be affected by a variety of factors, including a direct coupled relationship with temperature and a direct relationship with other actuators. With ventilation, a strategic approach in relation to external humidity is required. Equations [5] and [6] were used to calculate the band value for humidity control.

The automatic humidity supply system is determined by the control program

(FarmOS/Welgrow); when the system is switched on, a signal is transmitted through the fog controller, and water and compressed air are introduced to the greenhouse through nozzles (low air nozzle, Samchang GreenTech, Gimhae City, Korea) via water pumps (PU-350M, Wilo, Dortmund, Germany) and an air compressor.

Table 9 The specifications of fogging actuators installed in greenhouse.

	Description	Value
Nozzle	Nozzle diameter	0.5mm + 0.7mm
	Spraying range	4~7 m
	Spraying rate	80 ~100 ml/min
	Air pressure	392.26~588.40 kPa
Pump (PU-350M, Wilo)	Voltage	220 VAC
	Rated power	350W
	Pump head	10 m
	Maximum pump capacity	9,300 L/hr

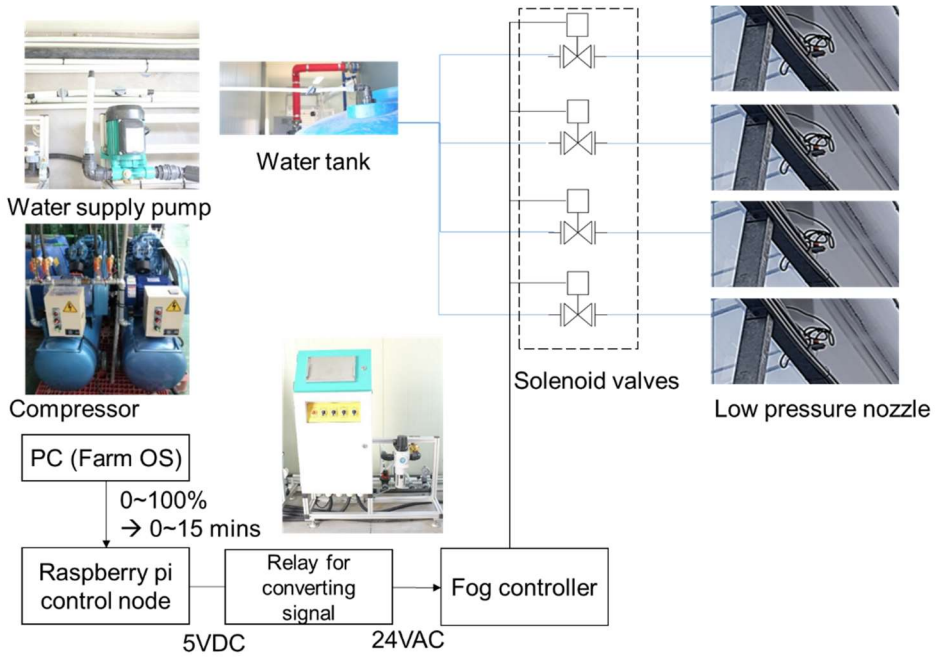

Figure 36. Overview of the automatic fog system used in the greenhouse.

$$\mathrm{PB} = X_1 + X_2 \cdot \frac{H_{om}}{H_{oc}} + X_3 \cdot \frac{T_{Im}}{T_c} \quad \cdots [5]$$

$$\mathrm{Output}\ (\%) = \frac{100}{PB} \cdot (E) \quad \cdots [6]$$

X_1–X_5: Influence coefficients

PB: Proportional bandwidth index

H_{om}: Maximum outside humidity (set value)

H_{oc}: Current outside humidity (measured value)

E: Difference between the current humidity and the target humidity

T_c: Current inside temperature

T_{IM}: Maximum inside temperature (set value)

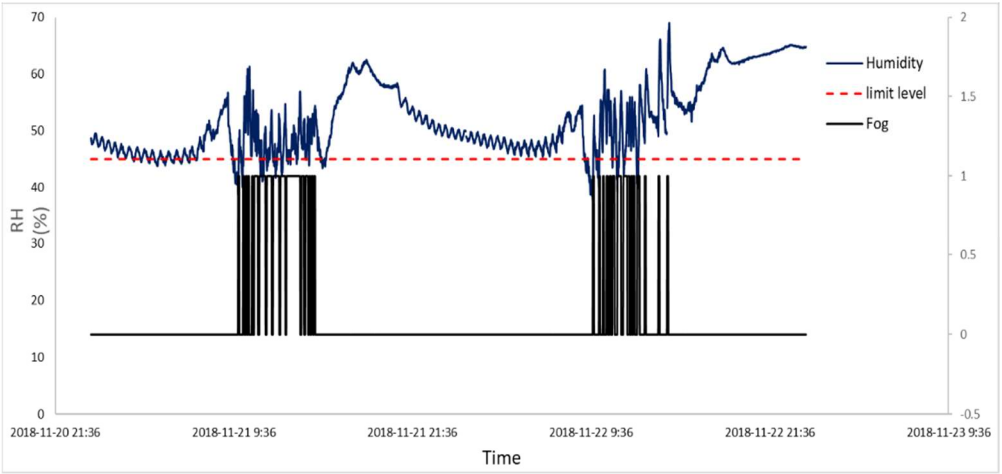

Figure 37. The result of the controlled humidity.

The relative humidity was controlled to remain at about 45% or below. As shown in Figure 37, outside dry air entered the greenhouse during the day, thus fogging control was implemented to prevent a drop in humidity. In the greenhouse, to improve control performance, a control design that considered various dynamic external disturbance factors was thus required.

3.2.4. HEAT RETENTION CURTAIN AND SHADE CURTAIN CONTROL

Shade and heat retention curtains were installed under the windows as shown in Figure 38, with the heat retention curtains under the shading curtains. The two curtain types had opposite opening and closing directions to each other, and two gear motors (COAM81062DA, Chung-oh Engineering, Daejeon City, Korea) were located in the center of the greenhouse. (Figure 38).

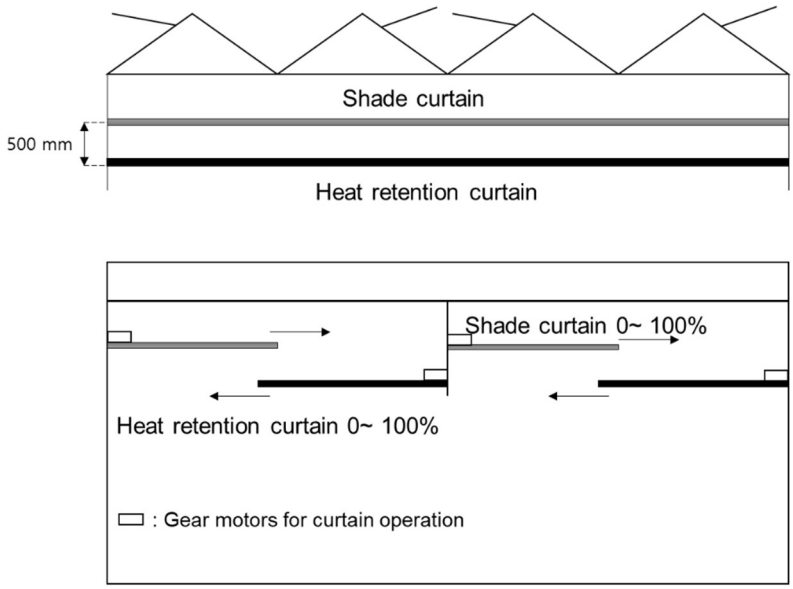

Figure 38. (A) Installation location and operating direction of the shading and heat retention curtains in the greenhouse and (B) their coverage area.

Table 10. Specifications for the gear motor installed to open and close the curtains in the greenhouse.

Description	Value
Model number	COAM 81062DA
Voltage	AC 380V (3 phase)
Rated current	0.65 A
Rated power	250 W
Motor speed	0.9 RPM
Reduction ratio	1:4000
Opening distance	100 m
Weight	10.2 kg

The performance of heat retention curtains depends on their thermal transmittance, which is the amount of heat that passes through the insulation material. The heat retention curtains used in the experimental greenhouse were a two-layer thermal insulation curtain combining a polyethylene film and an aluminum metalized film. The thermal transmittance was about 1.8 ($kcal \cdot m^{-2} \cdot hr^{-1} \cdot ℃^{-1}$).

Figure 39. Heat retention curtains installed in the greenhouse.

As the shade screen, an aluminum DTS 55 screen (DTS 55, Deltex, Yeoju City, Korea) was installed. The aluminum coating layer had fine pores so that water vapor could readily pass through (i.e., high moisture permeability).

Because the heat retention curtains were mainly closed at night, the temperature inside the greenhouse was used as a function to isolate the heat conduction with the outside of the curtain. In winter, the heat retention curtains can be used in conjunction with heating during the daytime, but the amount of radiation supplied to the crops is reduced by closing the curtains. On the other hand, in summer, it is generally not necessary to close the insulating curtains during the day or night. Equations [7] and [8] represent the P-band equations for the heat retention curtains, with the most important external factor being the outside temperature. Table 11 presents the initial coefficients set for heat retention curtain control.

$$\mathrm{PB}_{HC} = X_1 + X_2 \cdot \frac{R_m}{R_c} + X_3 \cdot \frac{T_{om}}{T_{oc}} \quad \cdots [7]$$

$$\text{Output } (\%) = \frac{100}{PB} \cdot E \quad \cdots [8]$$

X_1–X_3: Influence coefficients

PB_{HC}: Proportional bandwidth index for the heat retention curtain

R_m: Maximum solar radiation (set value, 1200 W/m^2)

R_c: Current solar radiation (measured value)

T_{om}: Maximum outside temperature (set value, 35 ℃)

T_{oc}: Current outside temperature (measured value)

E: Difference between the current temperature and the target temperature.

Table 11. Influence coefficients for each time slot for heat retention curtain control.

	X_1 (P base)	**X_2 (Radiation)**	**X_3 (Outside temperature)**
P_1	0	0	0
P_2	0	0	0
P_3	5.0	-3.0	-0.5
P_4	5.0	-4.0	-2.0
P_5	7.0	-3.0	-2.0
P_6	0	0	0

The shade curtains had similar driving conditions to the heat retention curtains but were made of a material with low heat insulation and high heat radiation. Although the sunshine was blocked, the curtains were less insulating than the heat retention curtains. During winter, the total amount of solar radiation was lower, thus the curtains were used at night to maintain the internal temperature in the same way as the heat retention curtains. During summer, the sunshine levels were very high, so the inner curtains were set to partially close when cooling of the greenhouse was required. Equations [9] and [10] represent the P-band equations for the shade curtains, and Table 12 summarizes the initial coefficients used for shade curtain control.

$$\mathrm{PB}_{SC} = X_1 + X_2 \cdot \frac{T_{om}}{T_{oc}} \quad \cdots [9]$$

$$\text{Output } (\%) = \frac{100}{PB} \cdot E \quad \cdots [10]$$

X_1 and X_2: Influence coefficients

PB_{SC}: Proportional bandwidth index for the shade curtains

R_m: Maximum solar radiation (set value)

R_c: Current solar radiation (measured value)

E: Difference between the current radiation and the target radiation.

Table 12. Influence coefficients for each time slot for shade curtain control.

	X_1 (P base)	X_2 (Outside temperature)
P_1	0	0
P_2	0	0
P_3	3.0	-2.0
P_4	3.0	-2.0
P_5	4.0	-2.0
P_6	0	0

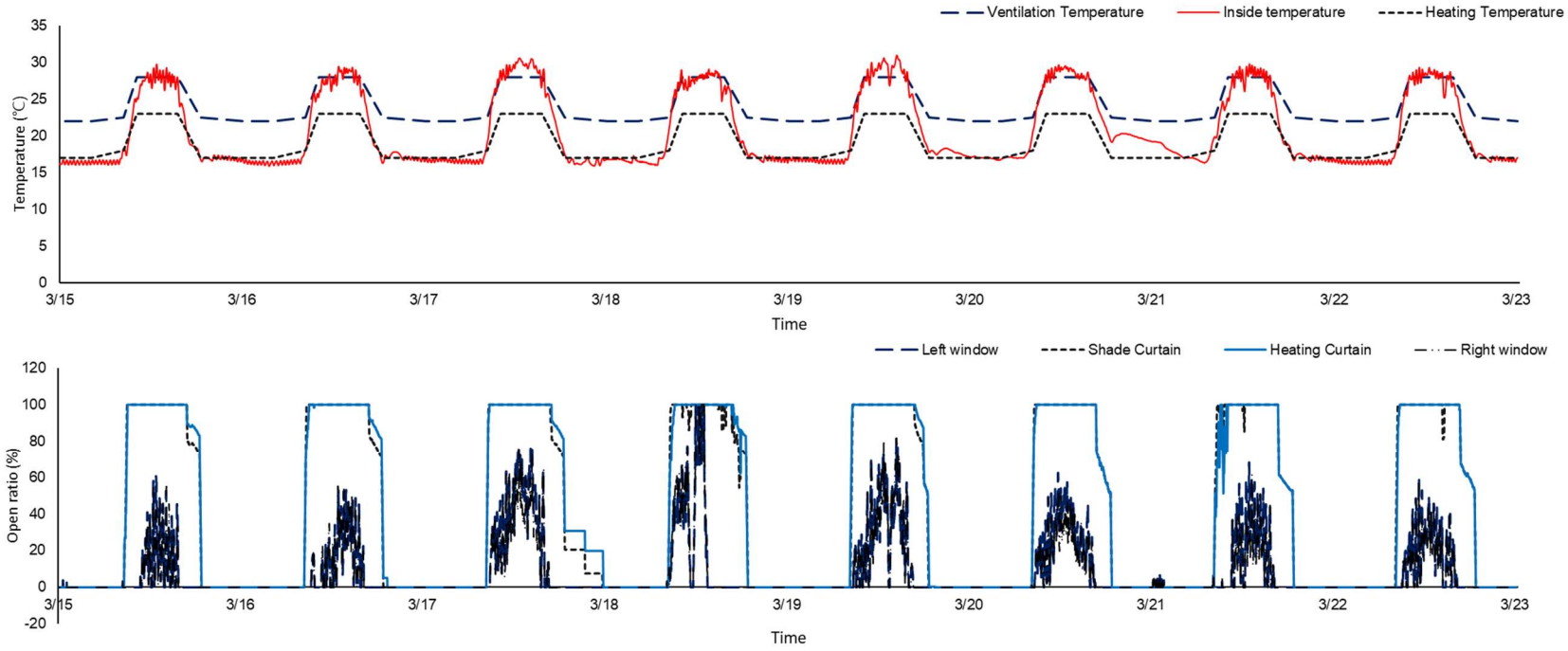

Figure 40. Controlled temperature by using P-band controller (top), and the opening ratios of windows and curtains(bottom) .

3.2.5. HEATING CONTROL

Water heating.

The majority of hot water heating used the cart rail for working carts inside the greenhouse at the same time as the heated pipes. High temperature (40−80°C) hot water flowed from a metal pipe to heat the pipelines, consequently heating the air inside the greenhouse. This type of heating can use a geothermal heat pump or a separate heating boiler to produce hot water. The hot water supply in this study was produced using a kerosene boiler (KDB-1035RTG, Navien, Seoul City, Korea).

The heater and the hot water supply were configured to operate automatically, and the controller generally sent two commands to the heating system: the target hot water temperature and the degree to which the linear-stroke valve was to be opened. The commands determined by the controller were the temperature of the hot water and the opening proportion of the linear-stroke valve actuator (ML7421A1032 E, Honeywell, North Carolina, United States). The degree to which the linear-stroke valve was opened regulated the supply of hot water through the rail pipe from the central heating pipe to the cultivation bed. Heating control was based on the target heating temperature set by the temperature control, and the opening proportion (%) of the linear-stroke valve was determined based on the outside weather, the solar radiation, and the deviation from the set value (Equations [11] and [12]). Table 13 presents the initial coefficients used for heating control.

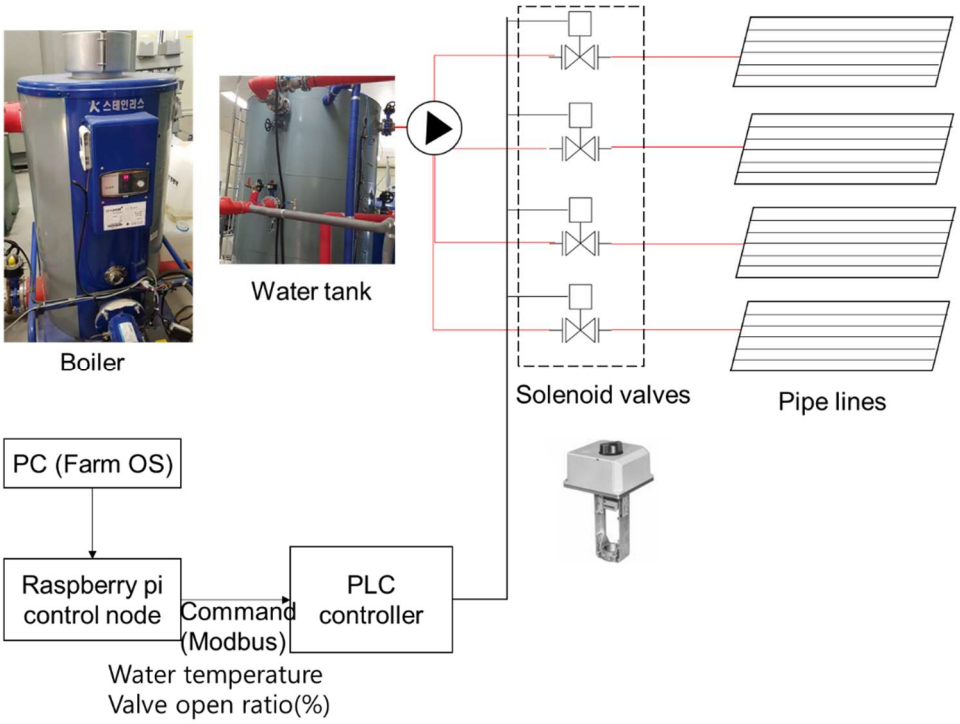

Figure 41. Overview of the automatic heating system used in the greenhouse.

$$\mathrm{PB}_H = X_1 + X_2 \cdot \frac{T_{om}}{T_{oc}} \quad \cdots [11]$$

$$\mathrm{Output}\ (\%) = \frac{100}{PB} \cdot (E + X_3 \cdot \frac{dE}{dt}) \quad \cdots [12]$$

X_1–X_3: Influence coefficients

PB_H: Proportional bandwidth index for heating

T_{om}: Maximum outside temperature (set value)

T_{oc}: Current outside temperature (measured value)

E: Difference between the current inside temperature and the target heating temperature.

Table 13. Influence coefficients for each time period for heating control.

	X_1 (P base)	X_2 (Outside temperature)	X_3 (D value)
P_1	4.0	0	-2.0
P_2	3.0	0	-2.0
P_3	10.0	3.0	-3.5
P_4	5.0	4.0	-2.0
P_5	7.0	3.0	-1.0
P_6	5.0	0	-1.0

Table 14. Specifications for the boiler installed for heating water.

Description	Value
Model number	KDB-1035RTG
Voltage	AC 220V (60 Hz)
Capacity of the hot-water supply	100,000 kcal/h (116.2 kW)
Rated thermal efficiency	91.3%
Fuel type	Kerosene (coal oil)
Fuel consumption rate	13.57 L/h
Heating area	4.2 m^2

3.3. EXPERIMENTAL RESULTS FOR THE OPTIMAL COEFFICIENTS OF VENTILATION CONTROL

3.3.1. THE DESIGN AND EXPERIMENT FOR THE RESPONSE SURFACE ANALYSIS METHOD TO OPTIMIZE VENTILATION CONDITIONS

The response surface analysis is suitable for fitting a quadratic surface and it helps to optimize the process parameters with a minimum number of experiments, as well as to analyze the interaction between the parameters (Betiku and Taiwo, 2015). Also, the characteristics of the response surface analysis are that information can be distributed across the entire experimental region, residual and prediction errors are

minimized, and a high level of examining compatibility lack is exhibited models can be designed sequentially, from a simple model to a high-order model. Also, the combination of factor levels used is minimized, so excellent graph analysis can be performed using simple data patterns. Minitab 17 (Eretec Inc., Korea) is a program which was used in this study. Minitab 17 is equipped with the necessary tools to prepare data for analysis and derive results through analysis. The response surface methodology analysis was conducted to optimize the five influence factors, X_1~X_5, for the P-band control. In this analysis, the central composite was designed in a grid pattern.

In this ventilation control experiment, the results for one sample was used as the response value by taking the RMSE difference of the temperatures collected during one day of greenhouse operation. The entire experiment lasted from April 2017 until May 2017. The spring weather condition was chosen because the reduction in temperature through ventilation is much needed during this period. Excluding the rainy days, the experiment was conducted over a total of 32 days. Also, a day was divided into six time slots, and the greenhouse control time slots were defined as P_1 through P_6. These time slots were reflected in the results, and the optimization coefficients for each time slot were compared (Table 4).

Table 15 Experimental design using central composite model of 32 trials

No.	X_1	X_2	X_3	X_4	X_5	No.	X_1	X_2	X_3	X_4	X_5
1	4	-0.5	-1.5	2	0.7	**17**	6	-0.5	-1.5	2	0.3
2	4	-1.5	-1.5	0	0.7	**18**	6	-1.5	-0.5	0	0.7
3	6	-0.5	-0.5	2	0.7	**19**	5	0	-1	1	0.5
4	5	-1	0	1	0.5	**20**	4	-0.5	-1.5	0	0.3
5	3	-1	-1	1	0.5	**21**	5	-1	-1	-1	0.5
6	6	-0.5	-0.5	0	0.3	**22**	5	-1	-1	3	0.5
7	4	-0.5	-0.5	2	0.3	**23**	5	-1	-1	1	0.9
8	4	-1.5	-0.5	0	0.3	**24**	5	-1	-1	1	0.5
9	5	-1	-1	1	0.5	**25**	5	-1	-1	1	0.5
10	5	-1	-2	1	0.5	**26**	6	-0.5	-1.5	0	0.7
11	6	-1.5	-0.5	2	0.3	**27**	6	-1.5	-1.5	0	0.3

12	7	-1	-1	1	0.5	**28**	5	-2	-1	1	0.5
13	4	-1.5	-0.5	2	0.7	**29**	5	-1	-1	1	0.5
14	5	-1	-1	1	0.5	**30**	6	-1.5	-1.5	2	0.7
15	5	-1	-1	1	0.5	**31**	4	-0.5	-0.5	0	0.7
16	5	-1	-1	1	0.1	**32**	4	-1.5	-1.5	2	0.3

Table 16 Time slots for the greenhouse climate control operation and the target temperature through ventilation for each time slot

	Time slot	Target ventilation temperature
P_1	~ 04:00	20
P_2	04:00 am ~ Sunrise	20-21
P_3	Sunrise ~ 4 hours after sunrise	21- 24
P_4	4 hours after sunrise ~ 4 hours before sunset	24- 28
P_5	4 hours before sunset~ Sunset	28- 24
P_6	Sunset ~ 24:00	24-20

3.3.1. APPLICATION RESULTS OF PD-BAND VENTILATION CONTROL

The PD-band ventilation control experiment was conducted over 32 days between June 15, 2017 and July 29, 2017—on days when it was not raining. The coefficient value was designed using the response surface analysis method, and it was provided as an input each day. The 32 sample data in were divided into 6 equal parts, and the RMSE value between the target ventilation temperature and the actual temperature for each time slot was used as the response value. The target ventilation values were divided into linear intervals for each time slot and was compared with the measured value using 1:1 comparison to compute the RMSE values. These results can be found in Table 6. Also, according to the plan for the response surface analysis experiment, which was designed for the purpose of finding the optimal parameters for the

ventilation settings, characteristics of the following parameters were investigated under each specified condition: the basic band value (X_1), the coefficient of solar radiation (X_2), the coefficient of the outside temperature (X_3), the coefficient of the wind speed (X_4), and the coefficient of the differential error (X_5). The conditional change characteristics were examined based on the RMSE error as the response variable (Table 5).

3.3.2. RESULTS OF THE LINEAR REGRESSION MODEL

By assigning the experimental results as the objective function of the response surface regression equation, the regression equation of second order [9] for the RMSE for the target ventilation value was derived. This equation was calculated by separating the P_1~P_6 coefficients for each time slot and the results, and the calculation results are shown in Table 17. Values for 20 coefficients were obtained using the analysis of variance (ANOVA), and 95% reliability was required for the analysis ($p<0.05$).

The data in Table 17. P_1~P_6 coefficients of the regression model of second order obtained through the response surface analysis method were analyzed using the response surface optimization method, and the results were displayed using 3-dimensional surface contour plots similar to ones shown in Figure 42. The figure shows the contour plots for the P4 time slot, which is when solar radiation is at the highest and the reduction in temperature is most needed due to the heat accumulated in the afternoon. This time slot is the most important period for ventilation control (Federico Villarreal-Guerrero et al., 2012). Here, the correlation between two influence factors can be verified. X_2 and X_3 are coefficients for the outdoor solar radiation and outside temperature, and it is shown that the more negative the influence of the two factors are, the lower the RMSE distribution is. This result can be attributed to the fact that the bandwidth is reduced in this period to increase how much the ventilation windows are opened because the temperature is more often higher than the target ventilation temperature during the P_4 time slot. This pattern is also shown in the graph of the correlation between solar radiation (X_2) and the coefficient of the differential error (X_5), as well as in the graph of the correlation between the outside temperature (X_3) and the bandwidth constant (X_1). For the

coefficient of wind speed (X_4), correlation with all other influence coefficients yielded a second order polynomial relationship.

$$RMSE(Px) = a + b_1 \cdot X_1 + b_2 \cdot X_2 + b_3 \cdot X_3 + b_4 \cdot X_4 + b_5 \cdot X_5 + b_6 \cdot {X_1}^2 + b_7 \cdot {X_2}^2 + b_8 \cdot {X_3}^2 + b_9 \cdot {X_4}^2 + b_{10} \cdot {X_5}^2 + b_{11} \cdot X_1 \cdot X_2 + b_{12} \cdot X_1 \cdot X_3 + b_{13} \cdot X_1 \cdot X_4 + \cdot b_{14} \cdot X_1 \cdot X_5 + b_{15} \cdot X_2 \cdot X_3 + b_{16} \cdot X_2 \cdot X_4 + b_{17} \cdot X_2 \cdot X_5 + b_{18} \cdot X_3 \cdot X_4 + b_{19} \cdot X_3 \cdot X_5 + b_{20} \cdot X_4 \cdot X_5 \quad \ldots [13]$$

Table 17. P_1~P_6 coefficients of the regression model of second order obtained through the response surface analysis method

No.	P_1	P_2	P_3	P_4	P_5	P_6	No.	P_1	P_2	P_3	P_4	P_5	P_6
1	2.83	1.31	2.11	2.14	2.87	1.09	**17**	1.01	1.66	2.99	1.79	3.27	1.35
2	3.00	1.46	1.20	2.32	2.68	1.25	**18**	0.72	1.62	0.23	1.95	2.22	0.80
3	1.28	1.46	1.98	1.82	2.26	0.40	**19**	1.86	2.07	2.24	2.79	3.15	1.27
4	1.95	1.98	2.11	2.53	3.10	1.56	**20**	2.92	2.27	2.37	2.81	3.57	1.99
5	3.56	1.95	2.31	2.73	3.48	1.89	**21**	1.36	2.27	0.92	2.41	3.03	1.49
6	1.08	2.46	1.78	2.80	3.50	1.77	**22**	2.27	0.94	2.81	0.28	2.64	1.25
7	3.13	2.02	3.05	2.77	3.62	2.17	**23**	2.14	1.37	0.98	2.09	1.92	1.16
8	2.80	1.96	2.14	2.73	3.46	2.17	**24**	1.80	1.60	1.73	2.17	2.79	1.49
9	1.57	1.72	1.73	2.07	2.94	1.36	**25**	2.06	1.56	1.72	2.13	2.92	1.51
10	1.97	1.38	1.74	1.71	2.63	1.05	**26**	0.51	1.78	0.54	2.04	2.34	1.22
11	0.81	1.45	2.77	0.89	2.98	1.94	**27**	0.25	1.67	1.14	1.85	2.96	1.61
12	0.35	1.41	1.44	1.42	2.30	0.95	**28**	1.65	1.19	1.50	1.38	2.60	1.50
13	2.96	1.17	1.94	1.12	2.75	1.64	**29**	2.07	1.71	1.96	2.17	2.94	1.32
14	2.03	1.69	1.81	2.21	3.05	1.48	**30**	1.03	0.59	1.41	0.00	1.59	0.64

15	2.10	1.56	1.96	2.13	3.03	1.35	**31**	2.63	2.38	2.04	3.20	3.31	1.51
16	1.44	2.04	2.87	2.31	3.79	2.30	**32**	2.54	1.22	2.36	1.05	3.06	2.11

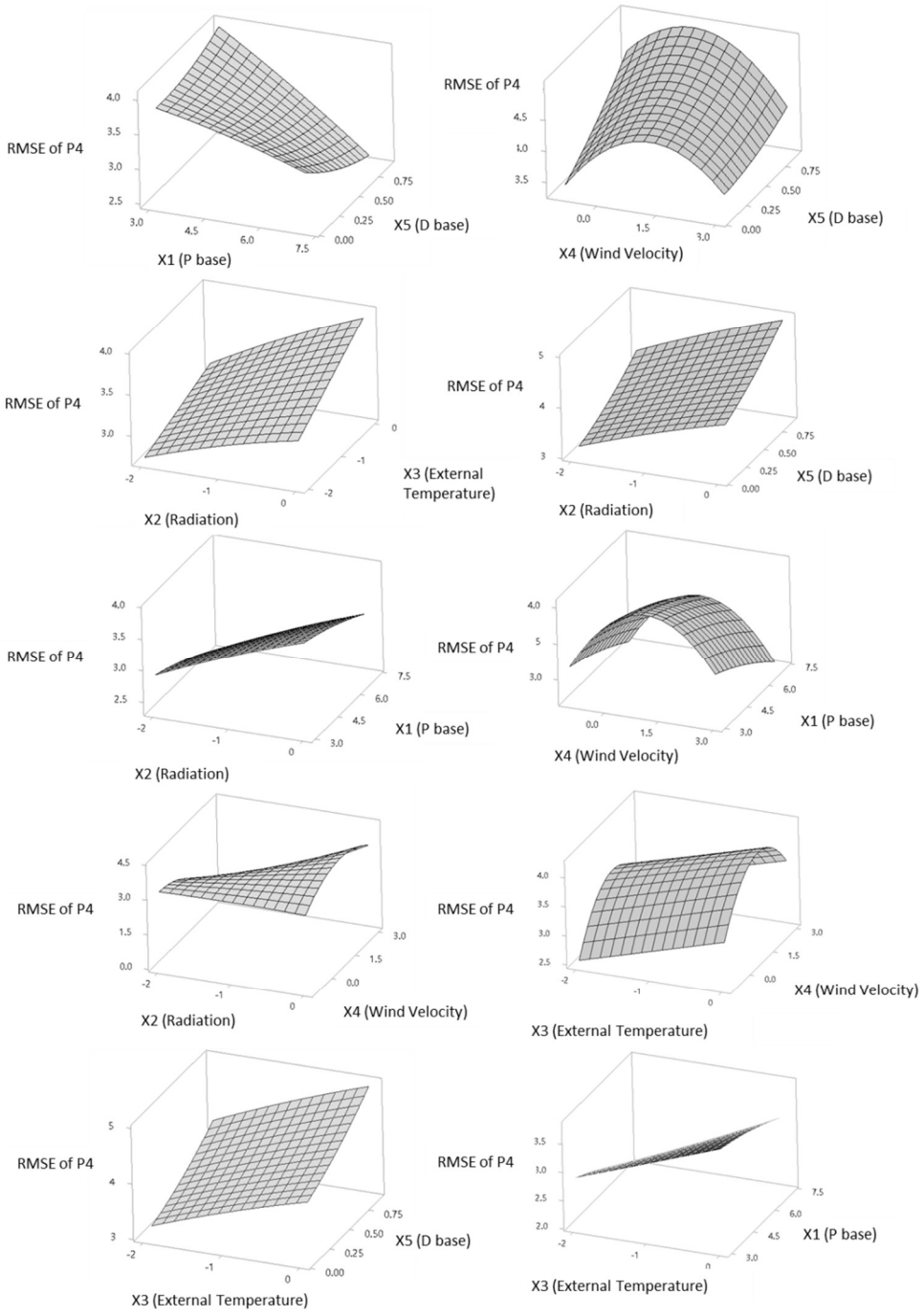

Figure 42. Response surface plots with two factors to RMSE of P4

3.3.3. RESULTS OF THE SURFACE RESPONSE ANALYSIS FOR OPTIMIZING THE COEFFICIENTS

The optimal values for the control influence coefficients (X_1~X_5) were obtained for each time slot (P_1~P_6) using the statistical numbers obtained through the surface response analysis method (Table 18). When the influence values for each time slot were examined, the influence values for the outdoor solar radiation and outside temperature is either zero or very small for P_1 and P_6 time slots. These time slots correspond to night time, so there is no outdoor solar radiation, and the outside temperature does not vary much. Also, the analysis results showed that the influence due to solar radiation (X_2) is negative. This result is attributed to the fact that the bandwidth is reduced to lower the temperature quickly because the greater the solar radiation value is, the greater is the effect of solar radiation on the rise of the indoor temperature. A similar phenomenon can be observed for the influence of the outside temperature (X_3). The influence of the wind speed (X_4) is primarily positive, but it doesn't appear to be a first order linear relationship as was verified by the 3-dimensional surface contour plots. The index for the influence (X_1) of the changes in the error exhibits a positive influence. Other than the P1 time slot, this value was 0.9 for all other time slots. This result indicates that three values for the influence coefficient—0.3, 0.5, 0.7—which were used in the previous experimental design, were rather low. The result also shows that responding to changes in the temperature deviation is an important element. In the existing climate controller, factor X_1 was non-existent. So it seems that low values were used during the experimental design because default values for this factor were not available. The coefficient values of X_1 to X_4 in the basic setup experiment were 5.0, 1.0, -1.0, and 1.0, respectively, and X_5 was not applied to implement common P-band logic.

Table 18 Optimal influence coefficients per time slot obtained through the surface response analysis method for optimizing ventilation control.

	X_1 (P base)	X_2 (Radiation)	X_3 (Outside temperature)	X_4 (Wind velocity)	X_5 (D-base)
P_1	5.66	0	0	-1	0.76
P_2	7.0	-1.12	-1.32	3.0	0.90
P_3	7.0	-2.0	0.029	1.12	0.90
P_4	7.0	-2.0	-2.0	0.78	0.90
P_5	7.0	-2.0	-2.0	1.28	0.90
P_6	6.77	-0.05	-0.05	0.98	0.90

The derived optimal coefficients were entered into the real system, and the operation of the greenhouse ventilation control was verified. The optimal coefficients were set for each time slot, and the ventilation control was performed to control the greenhouse temperature for three days. Figure 43 shows the temperature changes of the greenhouse before and after the application of the optimal coefficients. Figure 43 (top) shows the temperature change and the set temperature value, and the bottom of the graph shows the difference between the two values. For comparison, the temperature control performance through the optimal ventilation coefficients produced an RMSE value of 1.25 for the target temperature. When the experimental design method was conducted for approximately one month, the RMSE value of 2.03 was obtained for the average temperature. Therefore, in comparison, the 1.25 RMSE result verifies improved performance. In particular, the ventilation control performance was excellent in the time slots P_3 and P_5, when the temperature changed more drastically. As for the P_4 time slot, it seems there is a limitation in controlling the temperature through ventilation. For period such as this, a fogging method can be used to cool down the greenhouse.

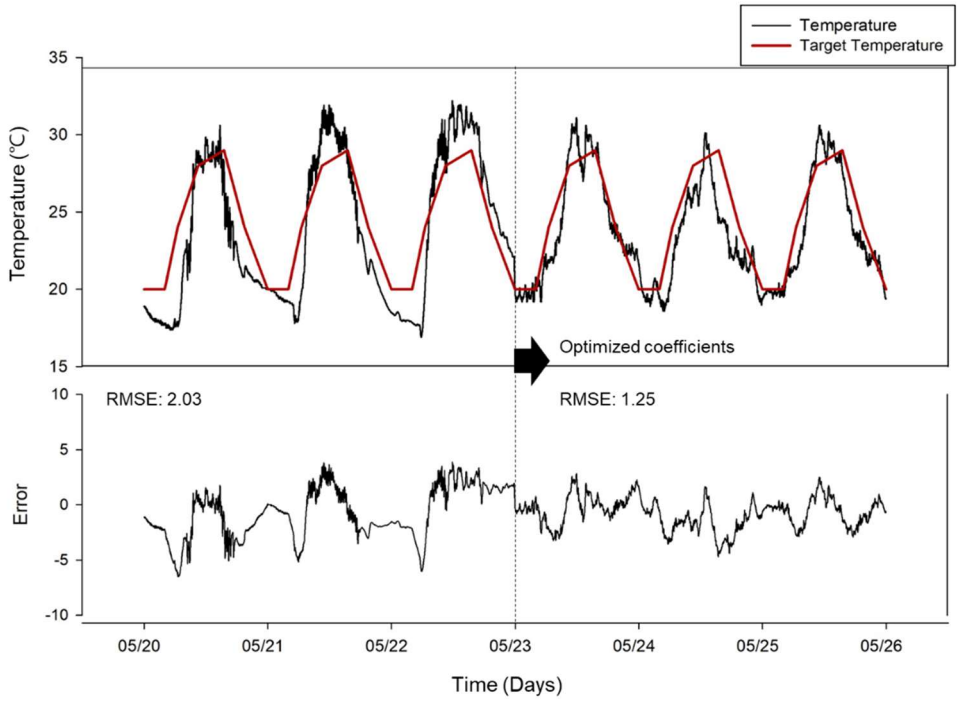

Figure 43. Temperature changes in the greenhouse managed using the optimal ventilation coefficients and the comparison with the target ventilation temperatures.

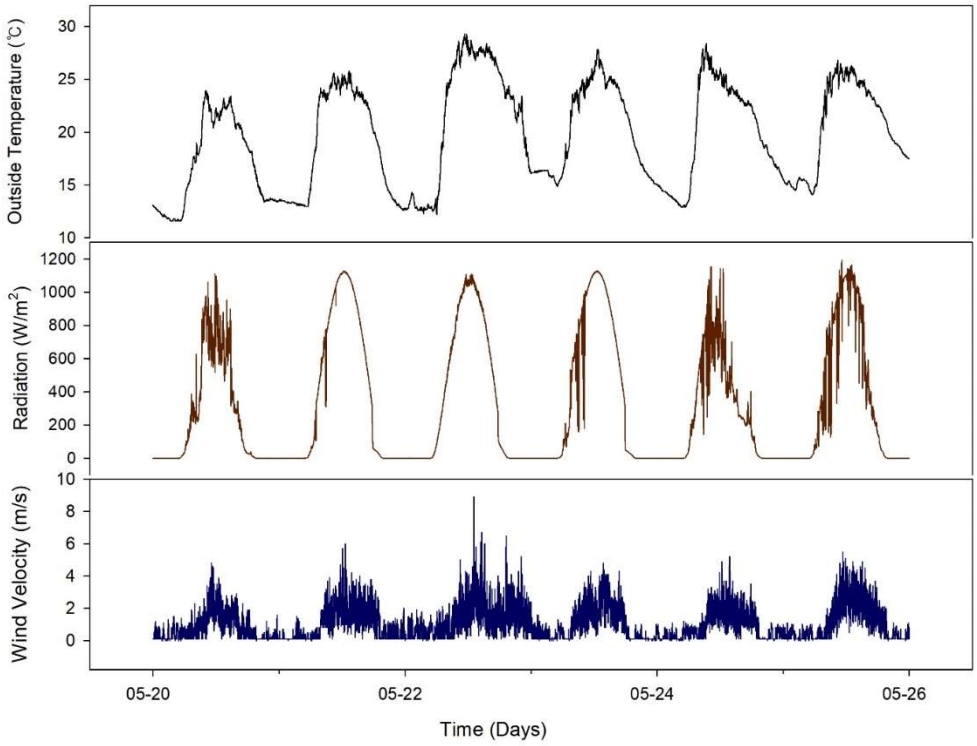

Figure 44. Changes in external environmental conditions in the experiment with optimal ventilation coefficients.

3.4. CHAPTER CONCLUSION

In this chapter, the structure of the smart greenhouse and the configuration of the actuators, sensors, and communication methods used in the experiment were described in detail. The control system used to collect meaningful information from the smart greenhouse was also outlined.

A linear algorithm-based (PD-band or P-band) control method was proposed to control the greenhouse actuators for the management of the temperature, humidity, and CO_2 levels, and its performance was verified. In ventilation control, the settings for the influencing factors were optimized for the PD-band, which determined the operation of the windows for ventilation. In addition, a P-band was applied when the actuator employed an on/off valve to consider complex factors in the control of the humidity and CO_2 concentration. The control results produced an RMSE of 14.45% and 35.44 ppm for the temperature and CO_2 concentration, respectively.

RSM analysis and an experimental statistical method were conducted. Based on the results, the conditions for optimal ventilation control were established. In order to optimize influential factors such as solar radiation, the external temperature, wind speed, the P-band width, and the D coefficient, which were required for PD-band ventilation control, 32 experimental conditions were designed. The values for each factor were applied to the actual operation of greenhouse ventilation, and the response values were obtained using the RMSE for the target ventilation temperature.

The greenhouse operation hours were divided into six time periods, and the response values were used to obtain the optimal coefficients for each time period. Using ANOVA analysis, a second-order polynomial equation was derived, and results were obtained for each time period. Finally, in order to minimize the RMSE, the optimal coefficients for the influential factors were calculated, and these values were applied to the actual greenhouse system. The ventilation control performance was evaluated, and the RMSE of 1.25 ℃ confirmed that the control performance had improved due to the use of the optimal coefficients. However, further long-term experiments need to be conducted in order to obtain the coefficients, and there is a concern that the optimization process here is specific to the greenhouse used in this experiment only.

It is also necessary to consider a method that can reduce the duration of the optimization process. This study proved that environmental settings, which are typically determined by the greenhouse operator's skills and experience, can be optimized using a statistical approach. It is anticipated that the study results will be useful in providing numerical guidelines for the factors influencing ventilation control settings. In addition, it can be very useful for the automatic control of appropriate settings based on external weather changes or different ventilation periods during the day. However, responding to nonlinear changes in the greenhouse environment with a linear control algorithm appears to have limitations. For this reason, efforts to interpret or control nonlinear motion using AI-based algorithms are required.

4. GREENHOUSE INSIDE CLIMATE PREDICTION BASED ON ARTIFICIAL INTELLIGENCE MODEL

A greenhouse is one of the major growing system by artificially manipulating a favorable environment for the plant. Especially maintaining adequate the temperature, humidity, and CO_2 concentrations have been major concerns to greenhouse environmental control because these environmental factors can improve plants development, quality, and quantity of plants produced. The greenhouse climate system considers as very complex and nonlinear system (El Ghoumari et al., 2005; Seginer and McClendon, 1992) in which the variables highly depend on the outside climate conditions and on the greenhouse design, and these climate conditions are unable to control independently (Fourati and Chtourou, 2007a; Frausto et al., 2003). Furthermore, Plants exposed to low or high temperatures or humidity may lead to mass death, dry plants due to low humidity and fungal diseases, which causes significant financial losses to growers. Therefore, building a precise model of greenhouse inside climate is important to the response of these dynamic changes and achieve an efficient climate management strategy (Fitz-Rodríguez et al., 2012; Yu et al., 2016a).

Lanfang et al., (2000) describes two different ways for building the models of greenhouse climate. One is based in terms of the physical laws involved in the process, and the other is based on an analysis of the input-output data of the process. Over the last decades, owing to increase of computational performance, numerous physical greenhouse methods have been presented (Benni et al., 2016a; Kishor and Singh, 2007; Norton et al., 2007; Soldatos et al., 2005). Nevertheless, this approach may yield inconsistent results when applying in real-world condition. This is because the models are defined by high complexity and often need to be calculated and estimated unmeasurable parameters such as, soil heat flux density, biological factors, photosynthesis rate, water vapor pressure and environment changing as well. Recently, the prediction methods based on data process have been proposed and applied for various fields due to development of the modern computational technology. This approach provides accurate results and fast process to implement agriculture field such as, greenhouse inside temperature prediction (Nury et al., 2017;

S L Patil et al., 2008), transpiration rate prediction (Wang et al., 2015) and ventilation strategy in livestock (Soldatos et al., 2005). In the various fields, the ANN models are powerful forecasting tools for analyzing the nonlinear system. Because of their ability to system models without the need to make any assumptions as are implicit in most traditional statistical approaches (Dariouchy et al., 2009). For this reason, these methods have been applied in the prediction of greenhouse climatic data and performed better results than the state of the physical models (Fitz-Rodríguez et al., 2012; He and Ma, 2010; Nury et al., 2017). In this method, however, some drawbacks have been raised on an optimization problem or an application issue in the real field, including over-fitting, need to many training sets, low versatile application to other plants, and sometime poor stability in strongly coupled and complex system.

Greenhouse inside temperature, humidity, and CO_2 concentrations are such strongly coupled factors by the influence of ventilation action, heating, fogging and other actuators and physical behaviors in the greenhouse. Moreover, the hindrance of making precise models is the facts that greenhouse has different actuators for the response each target climate and the most actuators have simple behavior on/off, and then regarded operation causes dominant changes of inside climate of the greenhouse. Several research has proposed a time-series model that should be able to provide reduced representations of large numerical systems for the accurate simulation and prediction of their dynamic response (El Ghoumari et al., 2005; Fan et al., 2012; Lu et al., 2014).

Since Lapedes and Farber (1987) reported the modeling of a nonlinear time series with an ANN, machine-learning algorithms that combine time series with regression models, such as the autoregressive (AR) model, the autoregressive moving average with exogenous inputs (ARMAX) model, and the nonlinear autoregressive with exogenous inputs (NARX) model, have received significant attention.

A recurrent neural network (RNN) is a special type of neural network that is designed for time series problems. Fourati and Chtourou (2007a) used an RNN based on an Elman structure and trained it to emulate the direct dynamics of a greenhouse temperature and hygrometry model. Recently, there have been significant improvements in the performance of RNNs, particularly with the introduction of

long short-term memory (LSTM) architecture to RNNs that is trained using backpropagation through time to overcome the vanishing gradient problem.

The aims of this chapter are to

1) Predict the inside temperature, relative humidity, and CO_2 concentration using external climatic data and historical data for actuators in a real greenhouse cultivating tomatoes
2) Build an ANN model for simultaneous forecasting and compare its performance in predicting the internal greenhouse climatic conditions with three time-series based models: NARX, an RNN model with LSTM (RNN-LSTM), and a convolutional neural network (CNN) with LSTM (CNN-LSTM)
3) Propose and evaluate an automatic model training and updating system based on an embedded board for the developed deep-learning models.

4.1. MACHINE LEARNING (ML) ESTIMATION MODEL

4.1.1. MACHINE LEARNING-BASED PREDICTION MODEL FOR INSIDE CLIMATE CHANGE OF GREENHOUSE

In general, the photosynthesis and evapotranspiration of crops are significantly affected by maintaining a constant temperature in a greenhouse. A constant temperature range is very important to the growth of crops. However, the temperature change in a greenhouse is impossible to predict. Greenhouses are frequently overheated by a large quantity of light in the daytime of summer. On the other hand, in winter, low outdoor temperatures cause the temperature loss in greenhouses, which results in excessive heating energy consumption.

The existing technique controls the environment of greenhouse by comparing the current temperature with a target temperature set by a user. Such a discreet control strategy is not effective in correcting or responding to a long-term temperature difference. Besides, single span greenhouse, which have been conventionally used to grow crops, are equipped with side and roof windows for ventilation. Since a linear algorithm is adopted to determine the opening ratios of windows in this

structure, an environmental change cannot be effectively reflected. Especially, in case a single span vinylhouse has a multiple number of windows, the inflow of outdoor air varies depending on whether side or roof windows are opened, and the indoor temperature may have a drastic change accordingly. For this reason, an optimal control is difficult to achieve by the existing method.

Although there have been many attempts to predict a temperature change inside a greenhouse by means of modeling, since necessary factors for modeling could not be measured, those attempts were not effective and failed to present accurate predictions. Recently, a learning algorithm of a nonlinear model was implemented for greenhouse modeling in order to derive parameters of PI or PID controller. However, this attempt aimed to determine a control logic, and accurate application was difficult due to a large error rate.

To solve the above problems, this study proposes an atmospheric environmental control system and the related control method, where two parallel neural networks are connected with each other. These two parallel neural networks are a prediction model, which predicts a greenhouse temperature after a certain time by applying empirical data to a machine learning-based training model, and an optimization model, which optimizes the control signals of a roof window by using a cost function concerning the difference from a target temperature.

4.1.2. ARTIFICIAL NEURAL NETWORK-BASED PREDICTION MODEL FOR GREENHOUSE INSIDE CLIMATE

Neural network model is usually applied in a forecasting non-linear system, whereas traditional time-series methods may not be able to capture the nonlinear pattern in data (Mitrea et al., 2009). The field of artificial neural networks is often just called Neural Networks or Multilayer Perceptrons after perhaps the most useful type of neural network. A Perceptron is a single neuron model that was a precursor to larger neural networks. It is a field of study that investigates how simple models of biological brains can be used to solve difficult computational tasks like the predictive modeling tasks in machine learning. The goal is not to create realistic models of the brain, but instead to develop robust algorithms and data structures that could use to model difficult problems. The performance of neural networks comes

from their ability to learn the representation in the training data and how to best relate it to the output variable. In this sense, neural networks learn a mapping. Mathematically, they are capable of learning any mapping function and have been proven to be a universal approximation algorithm. The predictive capability of neural networks comes from the hierarchical or multilayered structure of the networks. Neural networks can be applied to time series modeling without assuming a priori function forms of models. Many varieties of neural network techniques including an artificial NN, recurrent NN and Nonlinear Autoregressive eXogenous have been proposed and successfully applied to time series prediction (Fourati and Chtourou, 2007a; Moon et al., 2018; Sak et al., 2014). shows the structural differences of time-based learning algorithms.

Computational intelligence systems and among them, artificial neural networks (ANNs) has received attention recently in a field widely for approximation functions and forecasting (Altan Dombaycı and Gölcü, 2009; Khashei and Bijari, 2010; Patra, 1997). One of the most significant advantages of the ANN models over other classes of nonlinear models is that ANNs are universal approximates that can approximate a large class of functions with a high degree of accuracy. In the present study, we use an MLP trained with a gradient backpropagation algorithm to calculate the gradient of the error of each neuron network, the last to the first layer to predict the internal temperature, internal moisture inside CO_2 concentration of the greenhouse. The MLP consists of an input layer consisting of node(s) representing various input variable(s). The hidden layers consist of many hidden nodes and an output layer consisting of output variable(s). The input nodes pass on the input signal values to the nodes in the hidden layers. The values are distributed to all the nodes in the hidden layers depending on the connection weights between the input nodes and the hidden nodes. Activation functions can take several forms (Figure 46). The type of activation function is indicated by the situation of the neuron within the network. The most widely used activation function for the output layer is the linear function as non-linear activation function may introduce distortion to the predicted output (Khashei and Bijari, 2010).

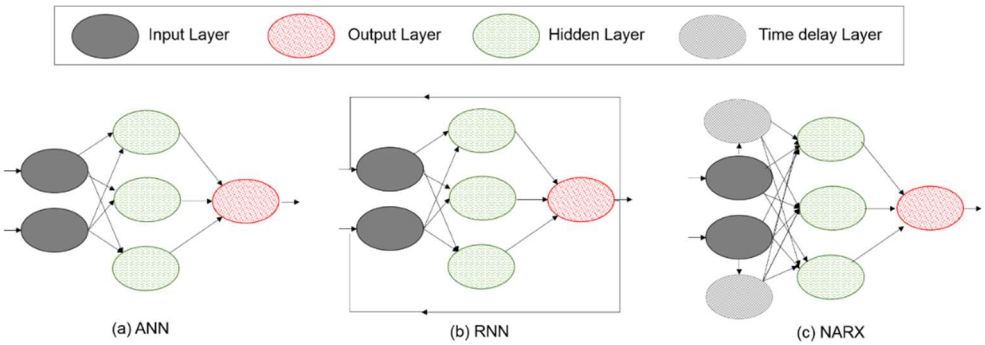

Figure 45. Structural comparisons of times series neural networks and traditional neural networks (Revised from Mitrea et al., 2009).

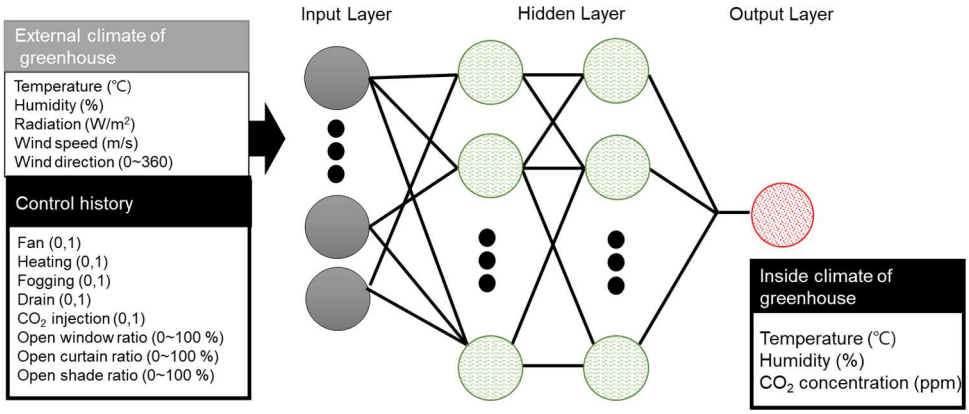

Figure 46. ANN basic Neuron for greenhouse climate prediction model.

4.1.3. GRADIENT DESCENT METHOD

The gradient of a multivariate function E (x_1, x_2, x_3 …x_n) is defined by Equation [14] below, which indicates the direction of the steepest increase of function value. As expressed in Equation [15], the gradient descent method determines the minimum value of a function by staring with an initial value (x_0=x_0, …, x_{n0}) and proceeding in the opposite direction to the gradient. Parameters for training neural network nodes and minimizing the value of cost function are gradually found.

$$\nabla E = \left(\frac{\partial E}{\partial X_1}, \frac{\partial E}{\partial X_2}, \ldots, \frac{\partial E}{\partial X_n}\right) \quad \cdots [14]$$

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k), k \geq 0 \quad \cdots [15]$$

The essence of the gradient descent method is moving from the current position toward the largest change of function value to obtain the maximum or minimum value of the function. Accordingly, if the direction of the largest change of function value can be obtained, the same concept will be applicable to various problems.

4.1.4. GAUSS-NEWTON METHOD

The Gauss-Newton method is a modification of the Newton method. This method is one of the representative optimization methods for nonlinear least squares problem. If the Newton method is applied to the optimization problem, the second order derivative is necessary. However, the Gauss-Newton method can find out a solution by means of the first order derivative.

If observed values are (x_i, y_i), i=1 ,.., n, model parameters are p = (p_1, p_2, ..., pm), a model is y = f(x, p), an error (residual) is $r_i(p) = y_i$ - f(xi,p) and the objective function to be minimized is E(p),as shown in the Equation [16].

$$\mathrm{E}(\boldsymbol{P}) = \sum_{i=1}^{n} r_i(\boldsymbol{P})^2 = [r_1(P) \cdots r_n(\boldsymbol{P})] \begin{bmatrix} r_1(\boldsymbol{P}) \\ \vdots \\ r_n(\boldsymbol{P}) \end{bmatrix} = r^T r \quad \cdots [16]$$

Here, the Gauss-Newton solution, which minimizes E(p) (that is, E'(p) = 0), is obtained by starting from the initial estimation $p_0 = (p_0, ..., p_m)$ for the model parameter p and iteratively updating p by means of the equations below. Jr is a simple expression of $Jr(p_k)$, which indicates the Jacobian matrix value of r in pk (Equation. [17] and [18]).

$$P_{k+1} = P_k - (J_r^T J_r)^{-1} \, J_r(P_K), \; k \geq 0 \qquad \cdots [17]$$

$$J_r(P) = \begin{bmatrix} \frac{\partial r_1}{\partial p_1}(P) & \cdots & \frac{\partial r_1}{\partial p_m}(P) \\ \vdots & \ddots & \vdots \\ \frac{\partial r_n}{\partial p_1}(P) & \cdots & \frac{\partial r_n}{\partial p_m}(P) \end{bmatrix} \qquad \cdots [18]$$

Equatin [18] is a simplified expression of Equation [19]

$$\mathbf{p}_{k+1} = \mathbf{p}_k - pinv(J_r)r(\mathbf{p}),\ k \geq 0 \qquad \cdots [19]$$

The essential principle of the Gauss-Newton method consists in finding a solution by locally approximating a nonlinear function to a linear function. If the error vector $r(p) = [r_1(p) \ldots r_n(p)]$ T is approximated to a linear function near pk by using the Taylor expansion, $r(p) \sim r(p_k) + Jr(p_k)(p - p_k)$ is obtained (the second or above order terms are neglected). When p is obtained, which minimizes the sum of squared errors $\| r(p_k)+Jr(p_k)(p-p_k) \| 2$ for a linearly approximated error, it can be expressed as $p = p_k - pinv(Jr(p_k))r(p_k)$ of Eq. (9). Consequently, the Gauss-Newton method obtains the least square solution (p_{k+1}) by linearly approximating an error function near the current parameter estimate (p_k) and linearly approximates the error function again near the solution to obtain the least square solution, thereby gradually approaching to the solution.

4.1.5. LEVENBERG-MARQUARDT METHOD

As mentioned above, the Levenberg–Marquardt method is a combination of the Gauss-Newton method and the gradient descent method. The gradient descent method is applied when the solution is far away. On the other hand, when the solution is near, the Gauss-Newton method is used to find it. However, the Levenberg–Marquardt method is more stable than the Gauss-Newton method in finding a solution (even if the initial value is far away from the solution, the Levenberg–Marquardt method is more likely to find it), and quickly converges on the solution. For this reason, the Levenberg–Marquardt method is applied to most of the nonlinear least square problems. The Levenberg–Marquardt method was proposed by Marquardt in 1963, who improved the Levenberg algorithm (1944) (Ranganathan, 2004). [20]

$$\mathbf{p}_{k+1} = \mathbf{p}_k - \left(J_r^T J_r + \mu_k\ diag(J_r^T J_r)\right)^{-1} J_r \mathbf{r}(\mathbf{p}_k),\ k \geq 0 \ \cdots [20]$$

The gradient descent method finds a solution (the minimum point where an error function is minimized) by moving in the opposite direction and by the step size

proportional to the size of gradient. The Levenberg method, which is an improvement of the Gauss-Newton method, aims to reduce the risk of divergence and find a solution more stably by adding the constant multiple μI of an identity matrix to $J_r^T J_r$. The constant μ(μ>0) is referred to as the damping factor. If μ has a small value, the Levenberg method becomes similar to the Gauss-Newton method. On the other hand, if μ had a large value, the Levenberg method becomes similar to the gradient descent method. If μ→∞, then $(J_r^T J_r+\mu_I)^{-1}\rightarrow 1/\mu_I$.. Accordingly, when μ increases, the Levenberg method is similar to the gradient descent method of which the step size is 1/μ. In the Levenberg method, however, the damping factor μ is not fixed but changes at each iteration (Figure 47). Accordingly, in the case of a stable convergence to a solution, μ is given a small value. Otherwise, that is, when the approach to a solution is not smooth, μ is given a large value. Specifically, if the error E(pk) calculated at the current step has been well decreased from that $E(p_{k-1})$ of the previous step, μk has a small value to find a solution through the Gauss-Newton method. On the other hand, if the error increases or the error decrease is not sufficient, μk needs to increase and the gradient descent method is applied to find a solution.

On the other hand, the Gauss-Newton method finds a solution by considering both gradient and curvature. In Equation [20], $J_r^T J_r$ has the significance of a Hessian, which a second order derivative (approximation matrix to Hessian). This indicates the curvature of a function. In other words, the step size for movement is determined by (the size of gradient)/ (the size of curvature). When the slope (gradient) is large but the curvature is also large (that is, the slope changes drastically), the minimum point is found by moving a little. On the other hand, if the curvature is small (that is, the slope hardly changes), the movement becomes larger to find a solution. Accordingly, the Gauss-Newton method can find a solution much more quickly and accurately than the gradient descent method. However, the Gauss-Newton method needs the calculation of the inverse matrix of $(J_r^T J_r)$. For this reason, in case $(J_r^T J_r)$ is approximate to a singular matrix (which has no inverse matrix), the calculated reverse matrix is numerically instable, which may result in divergence of solution.

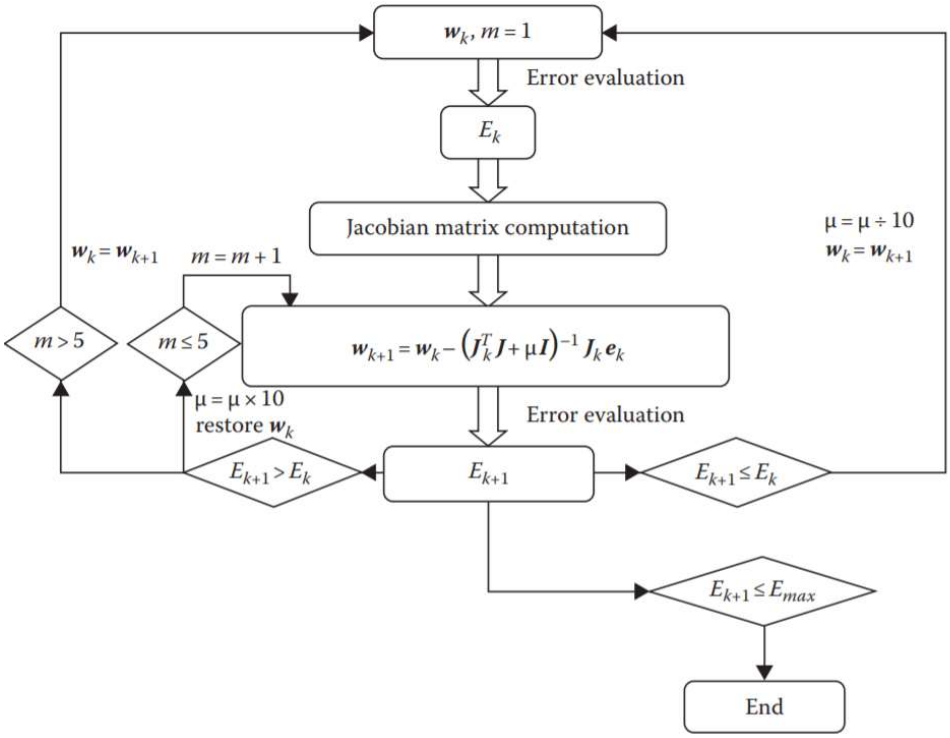

Figure 47. Block diagram for training using Levenberg–Marquardt algorithm: wk is the current weight, w_{k+1} is the next weight, E_{k+1} is the current total error, and Ek is the last total error.

Therefore, the training process using Levenberg□Marquardt algorithm could be designed as follows:

- With the initial weights (randomly generated), evaluate the total error (SSE).
- Do an update as directed by Equation 7 to adjust weights.
- With the new weights, evaluate the total error.
- If the current total error is increased as a result of the update, then retract the step (such as reset the weight vector to the precious value) and increase combination coefficient μ by a factor of 10 or by some other factors. Then go to step ii and try an update again.
- If the current total error is decreased as a result of the update, then accept the step (such as keep the new weight vector as the current one) and decrease the combination coefficient μ by a factor of 10 or by the same factor as step iv.

- Go to step ii with the new weights until the current total error is smaller than the required value.

4.2. TIME-SERIES BASED ALGORITHM MODEL

4.2.1. RECURRENT NEURAL NETWORK

There are various types of RNNs: Elman neural network, Jordan neural network, time lagged neural network, etc. This study mainly used a simple RNN that is basically similar to the most conventional feedforward neural network but has a recursive pathway, where the output of the middle layer unit is connected to itself.

The recurrent neural network (RNN) is any neural network with (directional) cyclic pathway. Such a structure of RNN enables the temporary memory of information and the corresponding dynamic change of reaction. Accordingly, the 'context' existing in a sequence of data is grasped and the above classification problem can be effectively solved. This is the most significant difference between RNN and the feedforward neural network.

There are various types of RNNs: Elman neural network, Jordan neural network, time lagged neural network, etc. This study mainly used a simple RNN that is basically similar to the most conventional feedforward neural network but has a recursive pathway, where the output of the middle layer unit is connected to itself (Figure 48).

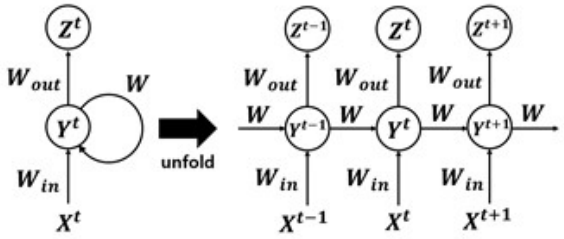

Figure 48. Basic 3-layered RNN model

Every RNN receives a single input xt at each time t and produces a single output y_t at the same time. When an output is calculated by a recursive pathway inside RNN, all the inputs received so far by the RNN are applied. The feedforward neural network models the mapping process where a single output is calculated for a single input. On the other hand, the modeling of RNN includes a different mapping process

where a single output is calculated for theoretically all the past inputs. The error function of RNN also sets a target (d_1, …, d_t) of an output sequence (y_1, …, y_t). This can be expressed as follows.

$$E(w) = -\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} d_n^t log y_k^t(x_n; w) \quad \cdots [21]$$

Because of the above characteristics, a little different method from backpropagation algorithms, which were used to train the conventional simple neural networks, is needed to train RNNs. This new method is referred to as BPTT (Backpropagation Through Time).

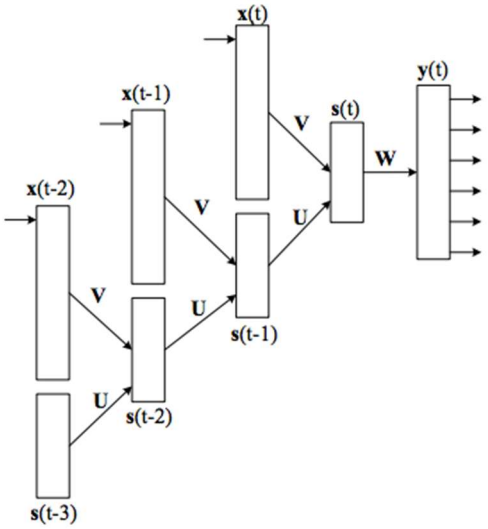

Figure 49. Weight connection flow of hidden layer (U, V, W) in Unfoloded RNNs

The weights of the above unfolded RNNs can be expressed as U, V and W below. To understand BPTT, the equation for the existing backpropagation (BP) algorithm needs to be applied to unfolded RNNs as shown in Figure 49. In order to obtain a value that is to be backpropagated, the error between a predicted value and a real value in the output layer is calculated as follows.

In BPTT, *W*, which is a weight of outputs in the hidden layer, can be updated in the same way as the conventional unfolded BP.

$$W(t+1) = W(t) + \eta s(t) e_o(t)^T \quad \cdots [22]$$

Here, the weight from the input layer to the hidden layer and the weight of the recursive pathway to the hidden layer calculate errors at a specific time, as follows.

$$e_h(t-\tau-1) = d_h(e_h(t-\tau)U, t-\tau-1) \quad \cdots [23]$$

When the sum of errors calculated for a certain time is backpropagated, the weight from the input layer to the hidden layer and the weight of the recursive pathway to the hidden layer are updated.

$$V(t+1) = V(t) + \eta \sum_{z=0}^{T} x(t-z) e_h(t-z)^T \quad \cdots [24]$$

$$U(t+1) = U(t) + \eta \sum_{z=0}^{T} s(t-z-1) e_h(t-z)^T \quad \cdots [25]$$

By the above update process, the error of the current time can be backpropagated to the past state for learning, as illustrated below Figure 50.

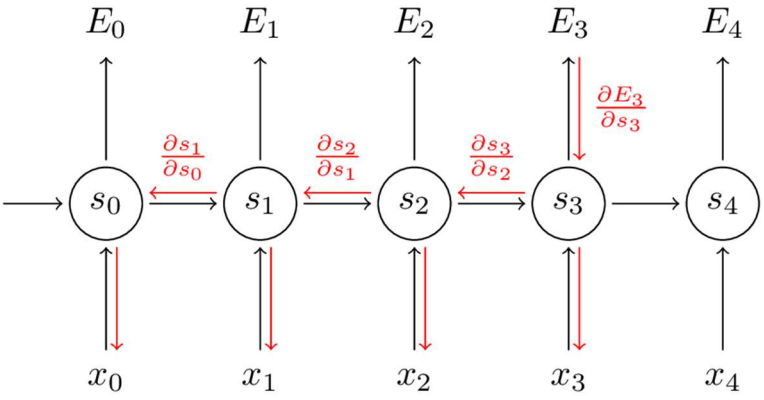

Figure 50. By the above update process, the error of the current time can be backpropagated to the past state for learning, as illustrated below.

4.2.2. LONG SHORT-TERM MEMORY

RNN can grasp and estimate the context of a data sequence, where the length of a context, which can be grasped, is essential. In other words, the important issue is how far the current input can be backpropagated. The required length of the context increases as the problem becomes more complicated. Theoretically, all the past inputs should be considered. However, 10 hours (This is argued on field applications)

is known to be the maximum possible length that can be reflected in RNN output. This limitation is attributable to the same cause as the gradient vanishing problem in the feedforward neural networks. In the case of a neural network with many layers, when the gradient is calculated by backpropagation, the calculation tends to drastically increase or vanish to zero while going back to layers. This can be understood in terms of the above-mentioned backpropagation of RNN. Even if RNN has a small number of layers, the backpropagation assumes many layers and thus the gradient is likely to either diverge or vanish. That is why RNN has a difficulty in dealing with a longer sequence. In other words, the above RNN can implement not a long-term memory but a short-term one. To solve this difficulty, some methods of realizing a long-term memory have been proposed. Among them, LSTM (Long Short-Term Memory) has been the most successful one (Hochreiter and Schmidhuber, 1997).

A RNN is a class of ANN where connections between nodes form a directed graph along a temporal sequence (Hochreiter and Schmidhuber, 1997). In this study, an RNN structure using time series of multivariate inputs was built, and output the results through the dense layer after passing through the LSTM (Figure 51). The dense layer normally consists of neurons, whose outputs are computed by using the hyperbolic tangent function as an activation function. The number of outputs in a dense layer is equal to the number of neurons. During the training process, the output values of the first dense layer are computed and passed to the next RNN layer. The RNN model developed using a tensor flow and Keras library based on Python 3.7.

The LSTM network consists of cells that take the input from the previous state and current input Xt. The past state, current memory, and present input work together to predict the next output. The main function of the cells is to determine the significance of the data, store it in memory, and transfer it to the next loop or ommit the memory. This capability allows the RNN to resolve a gradient vanishing in long time series analysis. The structure of the LSTM cell is shown in Figure 52; this operates in the following order. The first step is the forget gate (Equation. [26]), which receives information ht-1 and Xt outputs a number between 0 and 1 for each number in cell state Ct-1. An output number of 1 retains the data completely while 0 deletes the data.

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad \ldots [26]$$

The next step is divided into the update cells state and operate input gate, corresponding to Eqs. (18) and (19), respectively. The new cell state is determined by summing (a) the multiplication of the forget gate and the old cell and (b) the multiplication of the update cell and the input gate (Equation [27]).

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad \ldots [27]$$

$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \quad \ldots [28]$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad \ldots [29]$$

The output gate runs a sigmoid layer that decides the cell state (Equation. [30]), passes through the tanh function, and multiplies it by the output of the sigmoid gate (Equation [27]).

$$O_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad \ldots [30]$$

$$h_t = O_t * \tanh(C_t) \quad \ldots [31]$$

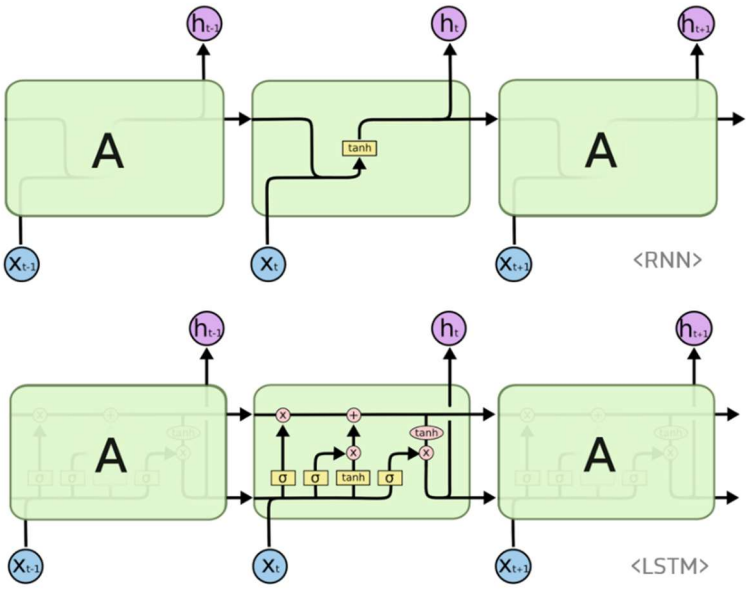

Figure 51. LSTM configured by adding a cell-state to the hidden state in RNN structure.

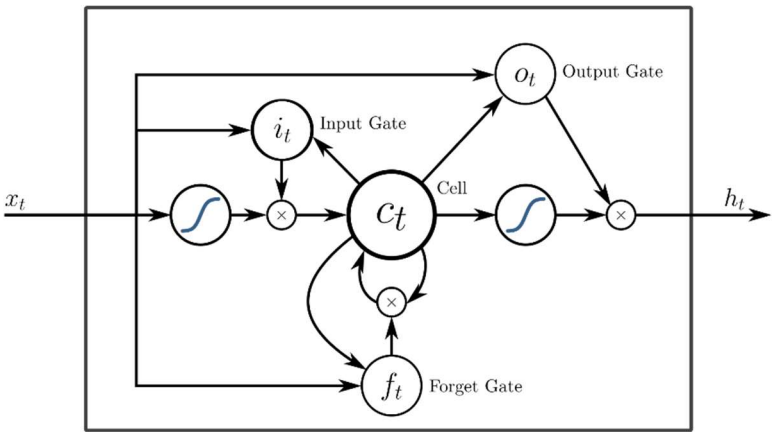

Figure 52. A LSTM unit with input (i.e.), output (i.e.), and forget (i.e.) gates. Each of these gates can be thought as a "standard" neuron in a feed-forward (or multi-layer) neural network.

4.2.3. NONLINEAR AUTOREGRESSIVE EXOGENOUS NARX

Dynamic networks are more powerful than static networks and generally somewhat more difficult to train. Because dynamic networks need to dispose of memory, they can be trained to learn sequential or time-varying patterns. NARX is one class of dynamic ANN models. The NARX model was developed based on autoregressive with exogenous input (ARX), which represents a linear system identification model. The NARX models have been widely applied in various fields because it can represent any nonlinear functions. (Menezes Jr and Barreto, 2006) proposed an architectural approach to deal with time series is one based upon "Nonlinear Autoregressive models with eXogenous input (NARX model), which is therefore called NARX neural networks (Diaconescu, 2008). All the specific dynamic networks discussed so far have either been focused networks, with the dynamics only at the input layer, or feedforward networks. NARX is a recurrent dynamic network, with feedback connections enclosing several layers of the network.(Figure 53). The NARX model is described on Equation. [32] which is based on the linear ARX model, which is commonly used in time-series modeling. The network was trained by an expanded Levenberg-Marquardt algorithm (with Bayesian regularization) which computes the new weights via the relationship.

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), u(t-2), \dots, u(t-n_u)) \dots [32]$$

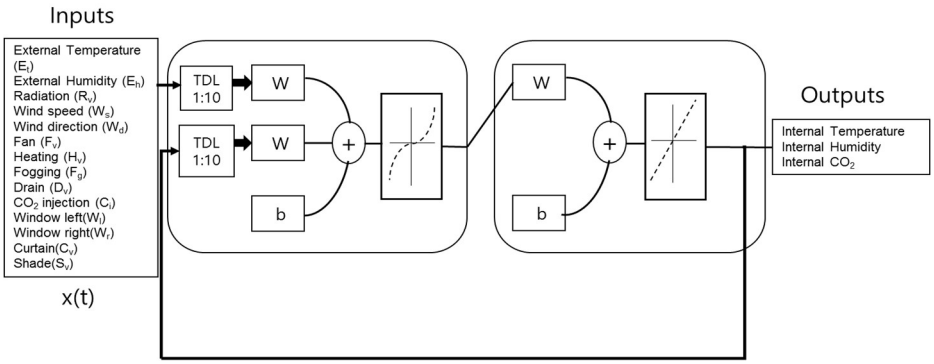

Figure 53. NARX structure for forecasting greenhouse climate value

4.3. DEVELOPMENT OF PREDICTION MODEL BY TIME STEP

The dataset used to develop the model included ~470,000 sample data points; these were prepared by outputting the target inside temperature, humidity, and CO_2 concentration with various time steps. In order to determine the predictability of changes in greenhouse inside temperature up to a few minutes, the data set was prepared every 5 min from 5−30 min because the determined operating time of the controller was < 5 min and > 30 min, so the model was developed with the structure shown in Figure 54 (a). Data for 3 days were separated and validated to compare the performances of the ANN and RNN-LSTM models. The prediction models' accuracy was assessed by substituting the outside weather condition and control history of actuators among the available training input variables (Table 26). The input data is normalized based on the maximum and minimum values to reduce the dependence between seasonal factors and input (Equation [33]).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \ldots[33]$$

Table 19. Ranges of input data used for building a prediction models

Input data (unit)	Range

Outside temperature (℃)	-4.5 ~ 34.8
Outside humidity (%)	29.5 ~ 100
Outside CO_2 concentration (ppm)	341.4 ~ 442.0
Radiation (W/m^2)	0 ~ 1311.2
Wind speed (m/s)	0 ~ 2.11
Wind direction (°)	0 ~ 359
Fan (0,1)	0,1
Heating (%)	0~100
Fogging (0,1)	0,1
CO_2 injecting(0,1)	0,1
Heat retention curtain (%)	0 ~100
Shade curtain (%)	0 ~ 100
Left window (%)	0 ~ 100
Right window (%)	0 ~ 100
Time	0 ~ 100

In an additional analysis, the prediction models were compared, when the input data was using only outside weather condition or the control history of actuators, respectively (Figure 54 (b)).

In addition, based on the developed environmental prediction accuracy, the results compared the model development performance by applying various training set conditions. In order to identify the minimum amount of training data required for model development, periods of 3, 6, 10, 20, and 30 d was used as training data sets based on the test set and for model development (Figure 54 (c)). The test data were randomly sampled from hot, cold, and warm external climate conditions. The average temperature for three climates was observed at 23.4 °C, 15.2 °C, and 2.4 °C for hot, warm, and cold climate, respectively.

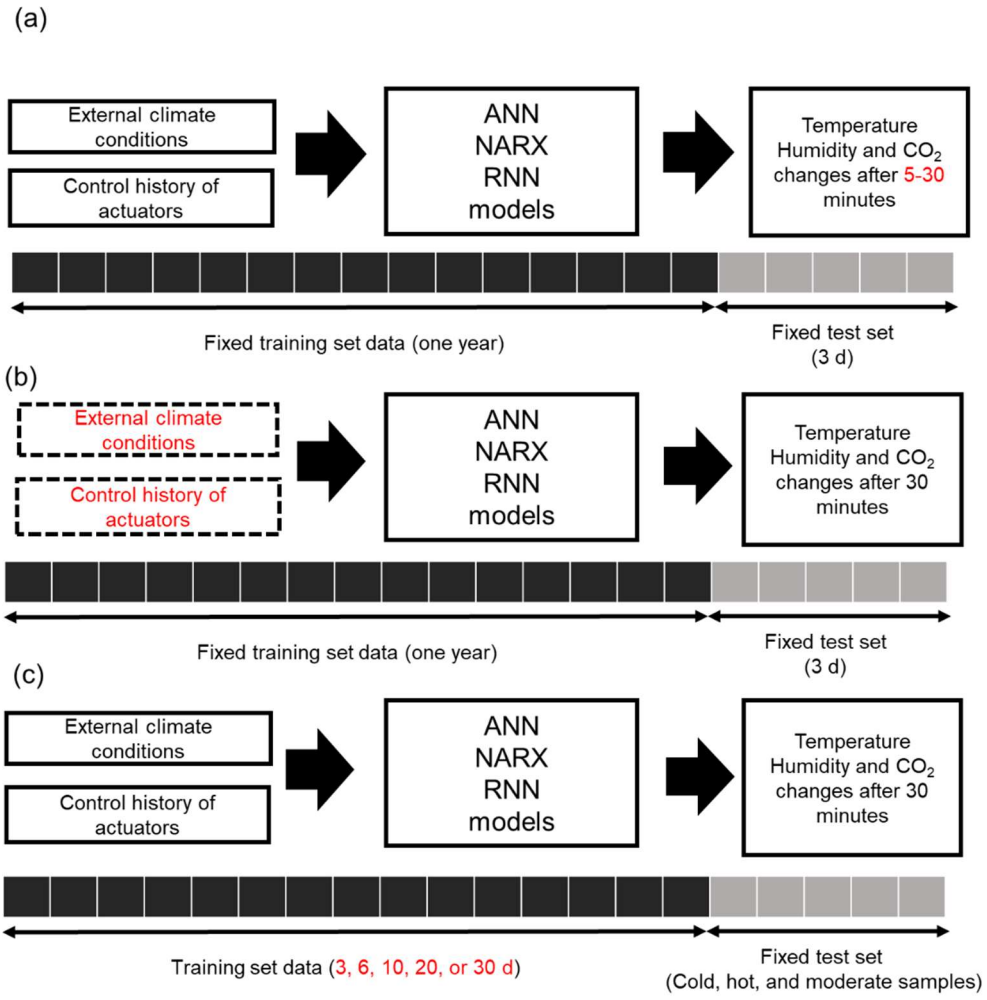

Figure 54. Schematic diagram of three approaches for building the prediction models (a): comparison of various time step, (b): model performance comparison of two categories of input variables, (c): model performance comparison by the number of training data set

A comparative analysis of the prediction models was conducted using the coefficient of determination (R^2, a measurement of the correlation between observed and predicted data) and the percent standard error of the prediction (% SEP) and the root mean square error of prediction (RMSEP), which determine how well the model explains the differences between predicted values and observed values (Andreas Kamilaris and Prenafeta-Boldú, 2018). (Equations. [34–36]):

$$RMSEP = \sqrt{\frac{1}{N}\sum_{i=1}^{n} (X_{obs,i} - X_{model,i})^2} \quad \ldots [34]$$

$$SEP\ (\%) = \frac{100}{\bar{X}_{obs,i}}\sqrt{\frac{1}{N}\sum_{i=1}^{n} (X_{obs,i} - X_{model,i})^2} \quad \ldots [35]$$

$$R^2 = 1 - \frac{SSE}{SSTO} = 1 - \frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{\sum_{i=1}^{n}(X_{obs,i} - \bar{X}_{obs,i})^2} \quad \ldots [36]$$

where SSTO a measurement of the variability of mean observed values, n is the total number of data sets used for estimation, $X_{obs,i}$ is the actual temperature (observed output), $X_{model,i}$ is the predicted temperature (estimated output), and $\bar{X}_{obs,i}$ is the mean value of the observed outputs of the prediction set. For a perfect match, R^2 should be close to 1 and the value of % SEP should be close to 0.

4.4. THE RESULTS OF TIME-SERIES PREDICTION MODELS

4.4.1. PREDICTION PERFORMANCE OF VARIOUS TIME STEP

Table 20 shows the accuracy results of the model predicted by ANN model. Temperature had the highest prediction accuracy. ANN-5 (prediction after 5 min) temperature had an RMSEP of 0.89 ℃, a SEP of 3.97%, and an R^2 of 0.94, as compared to humidity (5.67%, 8.59%, and 0.79, respectively) and CO_2 (31.23 ppm, 6.30 %, and 0.71, respectively); similar results were found for all time steps.

Table 20 Comparison of ANN prediction accuracy at time steps of 5, 10, 15, 20, 25, and 30 min for greenhouse temperature (°C), humidity (%), and CO_2 (ppm) change.

	ANN-5			ANN-10			ANN-15		
	Temp	Humidity	CO_2	Temp	Humidity	CO_2	Temp	Humidity	CO_2
RMSEP	0.89	5.67	31.23	0.92	5.66	30.23	0.82	5.67	32.18
SEP(%)	3.97	8.59	6.30	4.15	8.50	6.10	3.68	8.51	6.50
R^2	0.94	0.79	0.71	0.93	0.80	0.71	0.94	0.78	0.69
	ANN-20			ANN-25			ANN-30		
	Temp	Humidity	CO_2	Temp	Humidity	CO_2	Temp	Humidity	CO_2
RMSEP	0.87	5.10	30.45	0.90	5.23	31.23	0.94	5.44	32.12
SEP(%)	3.91	7.66	6.15	4.06	7.85	6.31	4.22	8.18	6.49

R^2	0.94	0.78	0.72	0.94	0.82	0.68	0.94	0.78	0.70

Table 9 and 10 show the results of applying NARX and RNN-LSTM model, respectively. In comparison, NARX's performance was lower than RNN overall, but slightly higher than ANN model within 5 to 15 min predictions. NARX-5 temperature had an RMSEP of 0.52 ℃, a SEP of 2.32%, and an R2 of 0.96. The NARX model has been observed to lose accuracy when the forecast time exceeds 15 min. In the case of RNN-LSTM model, the predictive performance was better than that of other models. RNN-5 temperature had an RMSEP of 0.45 °C, a SEP of 2.01% and an R2 of 0.97. The RNN-30 SEP was 3.15% (temperature), 7.85% (humidity) and 5.72% (CO_2).

Table 21 Comparison of NARX prediction accuracy at time steps of 5, 10, 15, 20, 25, and 30 min for greenhouse temperature (˚C), humidity (%), and CO_2 (ppm) change.

	NARX-5			NARX-10			NARX-15		
	Temp	Humidity	CO_2	Temp	Humidity	CO_2	Temp	Humidity	CO_2
RMSEP	0.52	3.01	19.24	0.58	4.20	22.27	0.88	5.61	33.65
SEP(%)	2.32	4.71	4.19	2.60	6.31	4.49	3.95	8.42	6.79
R^2	0.96	0.89	0.91	0.95	0.87	0.88	0.91	0.64	0.84
	NARX-20			NARX-25			NARX-30		
	Temp	Humidity	CO_2	Temp	Humidity	CO_2	Temp	Humidity	CO_2
RMSEP	1.27	5.33	32.55	1.12	5.21	32.52	1.32	6.27	28.30
SEP(%)	5.69	8.00	6.58	5.13	7.82	6.88	5.86	9.42	7.74
R^2	0.87	0.79	0.82	0.89	0.77	0.77	0.86	0.71	0.81

Table 22 Comparison of RNN-LSTM prediction accuracy at time steps of 5, 10, 15, 20, 25, and 30 min for greenhouse temperature (°C), humidity (%), and CO_2 (ppm) change.

	RNN-5			RNN-10			RNN-15		
	Temp	Humidity	CO_2	Temp	Humidity	CO_2	Temp	Humidity	CO_2
RMSEP	0.45	3.02	17.44	0.56	4.12	23.27	0.65	4.21	24.01

SEP(%)	2.01	4.72	3.80	2.51	6.19	4.70	2.92	6.32	4.85
R^2	0.97	0.88	0.85	0.96	0.82	0.80	0.96	0.81	0.83
		RNN-20			RNN-25			RNN-30	
	Temp	Humidity	CO_2	Temp	Humidity	CO_2	Temp	Humidity	CO_2
RMSEP	0.62	4.75	27.54	0.65	5.21	24.57	0.71	5.23	28.30
SEP(%)	2.78	7.13	5.56	2.98	7.82	4.90	3.15	7.85	5.72
R^2	0.96	0.86	0.83	0.96	0.81	0.82	0.96	0.80	0.81

Figure 55, and Figure *57* present a bar chart for the performance comparison of each model, and the prediction error of each model is compared by SEP. In addition, this comparison charts show the changes in the prediction error for each time step. Figure 55 shows the SEP changes for predicted temperature. The performance of ANN has not changed over time, but the accuracy of NARX and RNN-LSTM has decreased over time. For example, NARX became far less accurate than ANN after 20 min. In the humidity and CO_2 prediction model, time-based algorithms, NARX and RNN-LSTM, have been found to increase in error over time (Figure 55Figure 56 and Figure 57).

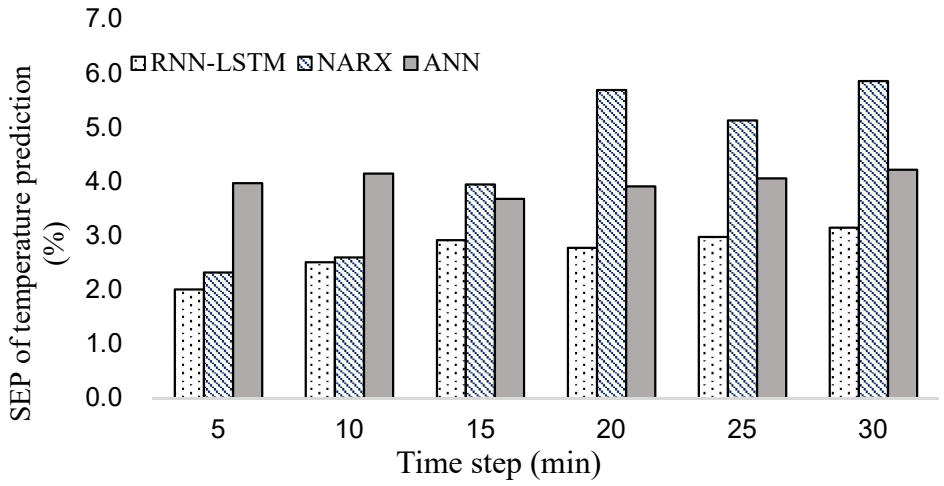

Figure 55. Comparison of SEP changes per time step for temperature prediction models

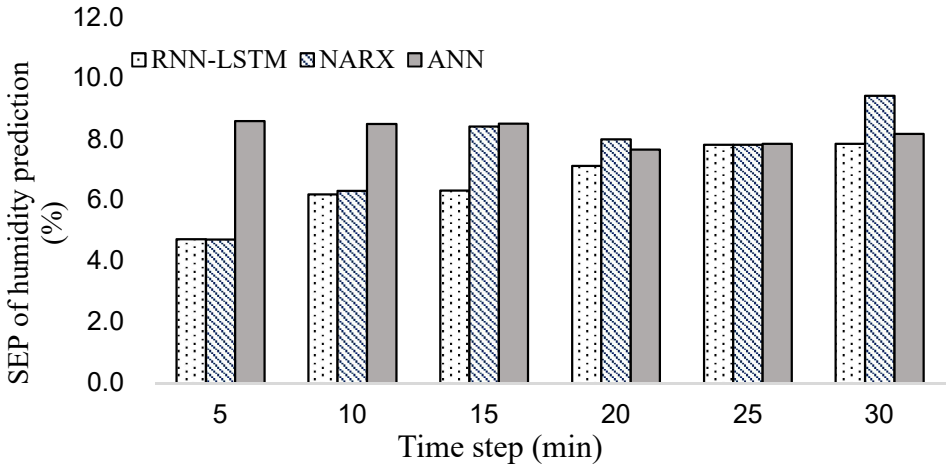

Figure 56. Comparison of SEP changes per time step for humidity prediction models

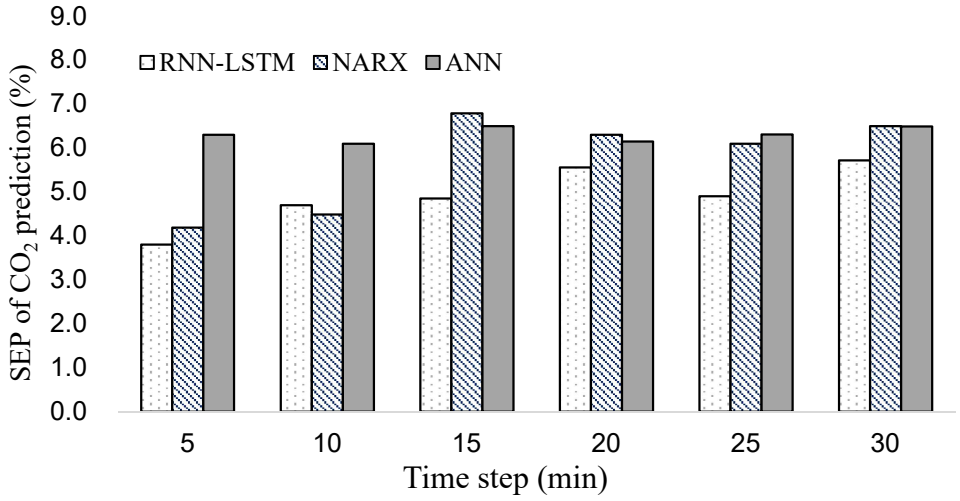

Figure 57. Comparison of SEP changes per time step for CO_2 prediction models

Table 23 shows the results obtained by classifying the training data sets by type and separately excluding different sets, using models forecasting changes after 30 min with a one-year training data set. The validation data for this test was the same as that used in Table 10. The greatest drop in accuracy occurred when the input factor corresponding to the current values of the targets was removed. For example, the tendency of RNN-LSTM to depend on the present value as compared with external weather conditions or control history was confirmed. Humidity prediction was the most challenging; RNN-LSTM had an R^2 of 0.80 when using all training data sets.

Table 23 Comparison of accuracy changes for each training set when external weather conditions or actuator control history, was used. Values in parentheses are the difference from results when using all datasets.

Temperature		All datasets	Excluding external weather conditions	Excluding actuator control history
ANN	SEP	4.22%	5.34% (+1.12%)	6.34% (+2.12%)
	R^2	0.94	0.91 (-0.03)	0.85 (-0.09)
NARX	SEP	5.86%	6.24% (+0.40%)	7.96% (+2.10%)
	R^2	0.86	0.83 (-0.03)	0.80 (-0.06)
RNN-LSTM	SEP	3.15%	3.47% (+0.32%)	4.39% (+1.24%)
	R^2	0.96	0.95 (-0.01)	0.90 (-0.05)
Humidity		ALL	Excluding external weather conditions	Excluding actuator control history
ANN	SEP	8.18%	9.20 % (+1.02%)	10.86 % (+1.68%)
	R^2	0.78	0.74 (-0.04)	0.71 (-0.07)
NARX	SEP	9.42%	9.49 % (+0.07%)	10.84 % (+1.06%)
	R^2	0.71	0.70 (-0.01)	0.62 (-0.09)
RNN-LSTM	SEP	7.85%	7.89 % (+0.04%)	8.86 % (+1.01%)
	R^2	0.80	0.79 (-0.01)	0.77 (-0.3)
CO_2		ALL	Excluding external weather conditions	Excluding actuator control history
ANN	SEP	6.49%	8.26% (+1.79%)	7.89% (+1.40%)
	R^2	0.70	0.67 (-0.03)	0.67 (-0.03)
NARX	SEP	7.74%	7.81% (+0.70%)	8.67% (+0.93%)
	R^2	0.81	0.77 (-0.04)	0.72 (-0.09)
RNN-LSTM	SEP	6.79%	7.01% (+0.32%)	7.97 % (+1.18%)
	R^2	0.84	0.81 (-0.03)	0.80 (-0.04)

4.4.2. COMPARISON OF VALIDATION RESULTS FOR THE THREE MODELS IN VARIOUS TRAINING CONDITIONS

Figure 58, Figure 59, and Figure 60 show the prediction accuracy of a particular test sample over various learning sets for ANN, NARX, and RNN-LSTM, respectively, during varying climate conditions and number of training days. Overall, all three models showed relatively high accuracy for temperature prediction during the three climate conditions.

ANN using 3 d of training data produced SEPs of 13.12% (temperature), 17.97% humidity), and 12.65% (CO_2). After 30 d, the performance for temperature was significantly improved (8.83%) but still poor for humidity (14.29%) and CO_2 (12.37%). The NARX temperature prediction had good prediction performance even with a relatively small data set (SEP of 4–9%). CO_2 performance improved with training days, reaching 5.37% SEP for 10 d, humidity remained poor throughout with a lowest SEP of 12.29%. RNN-LSTM had the best overall prediction performance of the three models. For example, using 10 d of training data, prediction was excellent with SEPs of 4.07% (temperature), 7.17% (humidity), and 5.35% (CO_2). Using 30 d, the temperature SEP was reduced to 3.83%, a highly satisfactory performance, while humidity reached 5.29%, far better than the other models.

In addition, the climate prediction model to be applied to greenhouses needs to be studied in a structure that includes image data of crops in the future. Alternatively, it is necessary to add information about the growth of crops by reflecting the index such as additional accumulated days after planting (DAP) of the input.

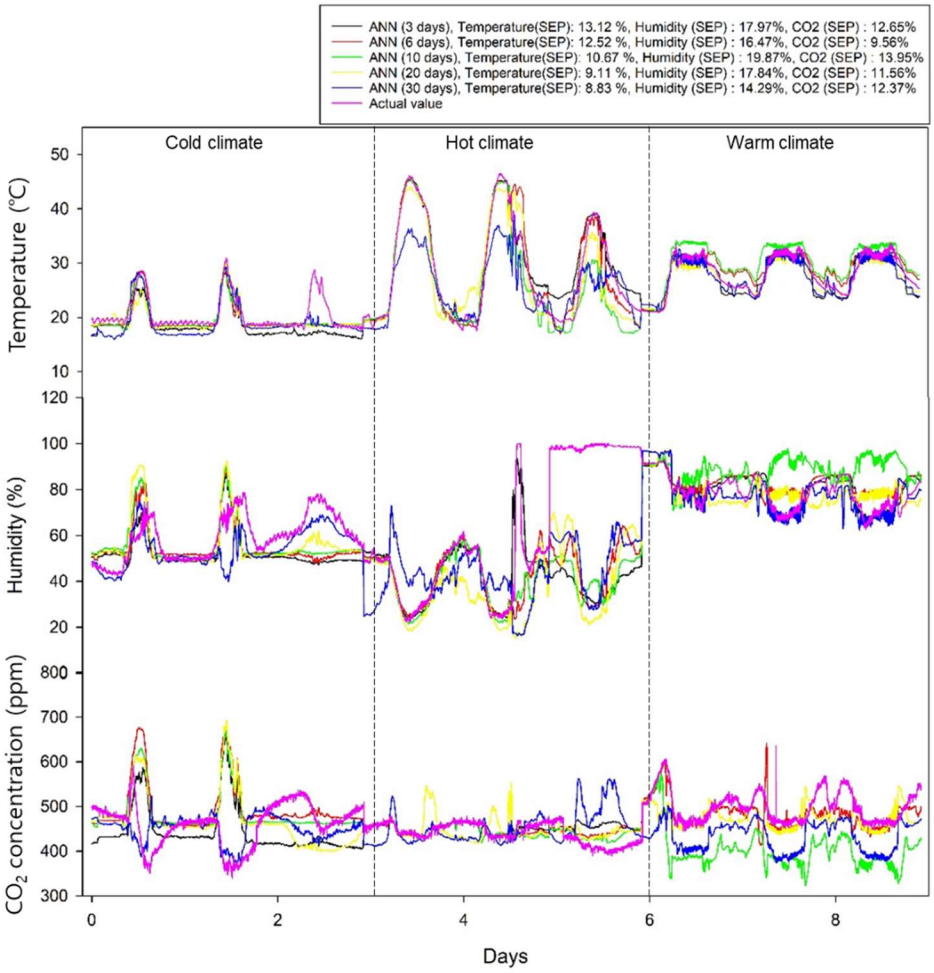

Figure 58. ANN prediction results using a test set of cold (days 1–3), hot (days 4–6), and moderate (days 7–9) climates with various training datasets.

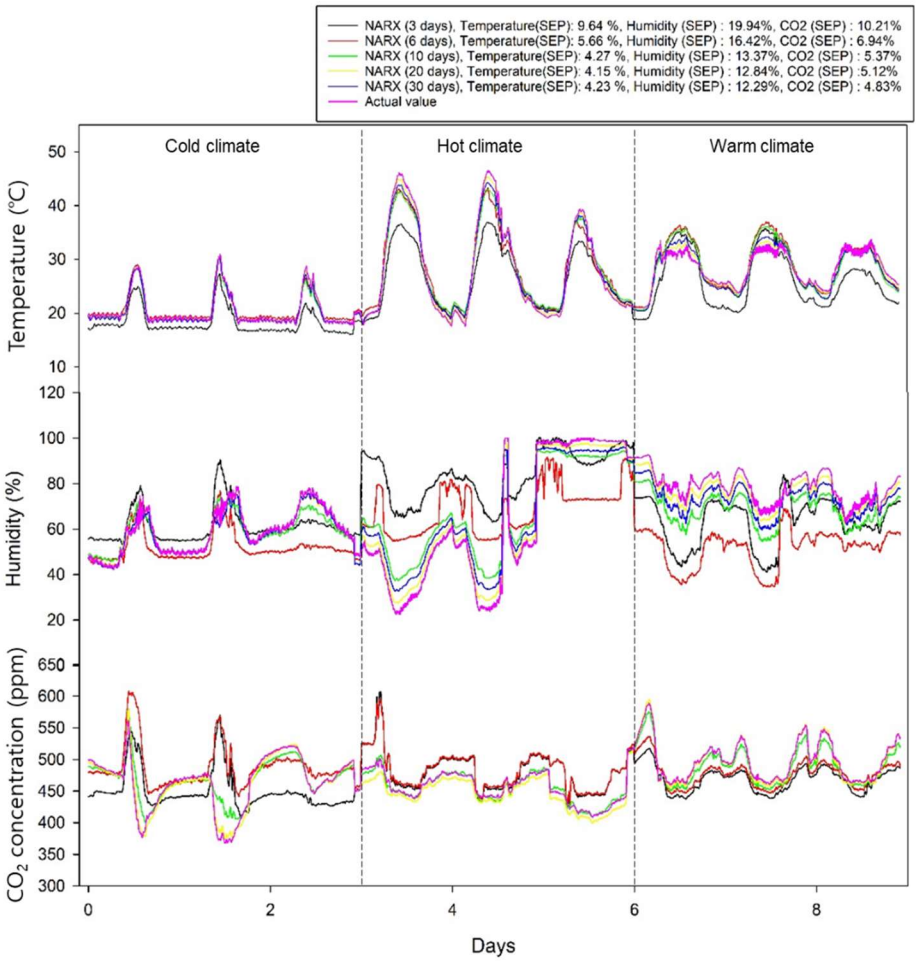

Figure 59. NARX prediction results using a test set of cold (days 1−3), hot (days 4−6), and moderate (days 7−9) climates with various training datasets.

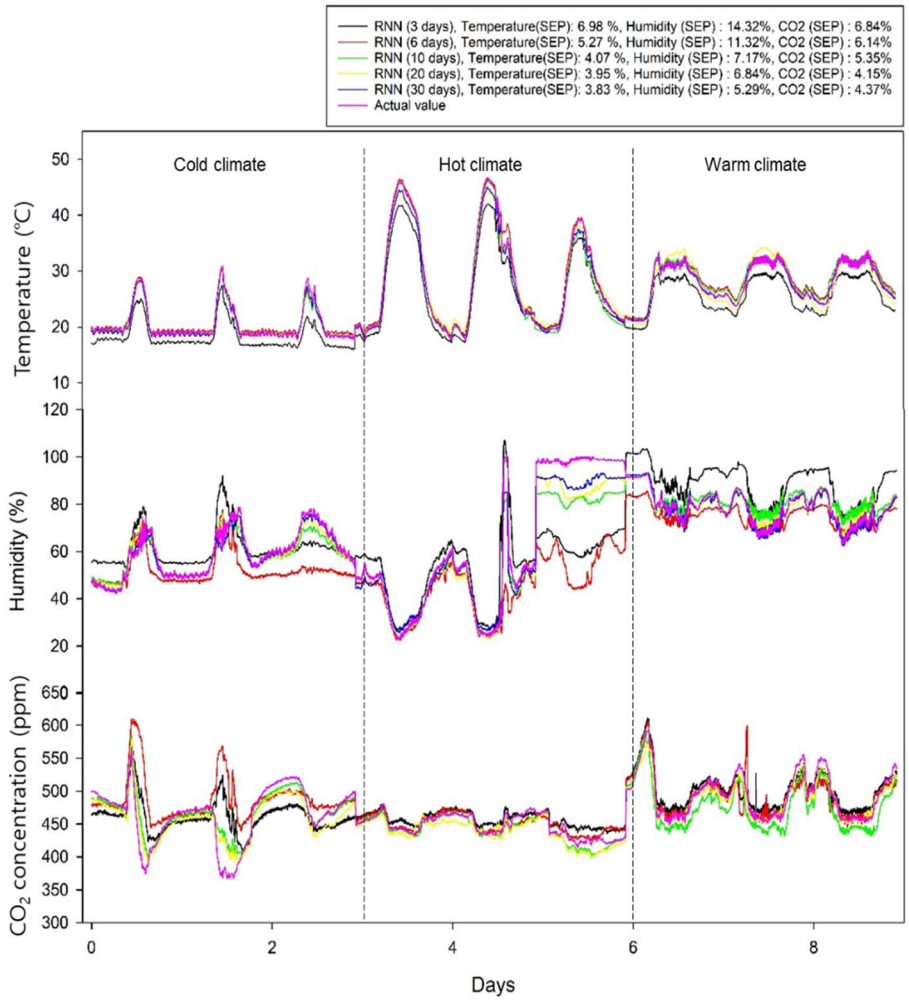

Figure 60. RNN-LSTM prediction results using a test set of cold (days 1−3), hot (days 4−6), and moderate (days 7−9) climates with various training datasets.

4.5. CONVOLUTIONAL NEURAL NETWORK (CNN)-LSTM MODELS FOR CLIMATE PREDICTION

CNNs have recently been employed in agriculture to address various agricultural problems with increasing success, with more than 20 studies employing CNNs having been reported to date (Grinblat et al., 2016; Kamilaris and Prenafeta-Boldú, 2018). Because CNNs may be the most widely used technique in agricultural research today for problems related to image analysis, this current review focuses on this specific sub-set of deep-learning (DL) models. To the authors' knowledge, this is the first review for the agricultural domain that focuses on CNNs, although a small number of more general studies do exist (Andreas Kamilaris and Prenafeta-Boldú, 2018; LeCun et al., 2006) that present and analyze related work in other research domains and application areas.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons are typically fully connected networks in which each neuron in one layer is connected to all of the neurons in the next layer. This structure makes them prone to overfitting the data. Typical methods of regularization include adding some form of weight to the loss function, but CNNs take a different approach: they take advantage of hierarchical patterns in data and assemble more complex patterns using smaller and simpler patterns. Therefore, in terms of connectedness and complexity, CNNs are on the lower extreme.

CNNs were inspired by biological processes in that the connectivity patterns between neurons resemble the organization of an animal's visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms are hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

4.5.1. CONVOLUTIONAL LAYER

A CNN consists of an input and an output layer, as well as multiple hidden layers (Figure 61). The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or another dot product. The activation function is commonly a RELU layer, which is subsequently followed by additional convolutions such as pooling layers, fully connected layers, and normalization layers, collectively referred to as hidden layers because their inputs and outputs are masked by the activation function and the final convolution.

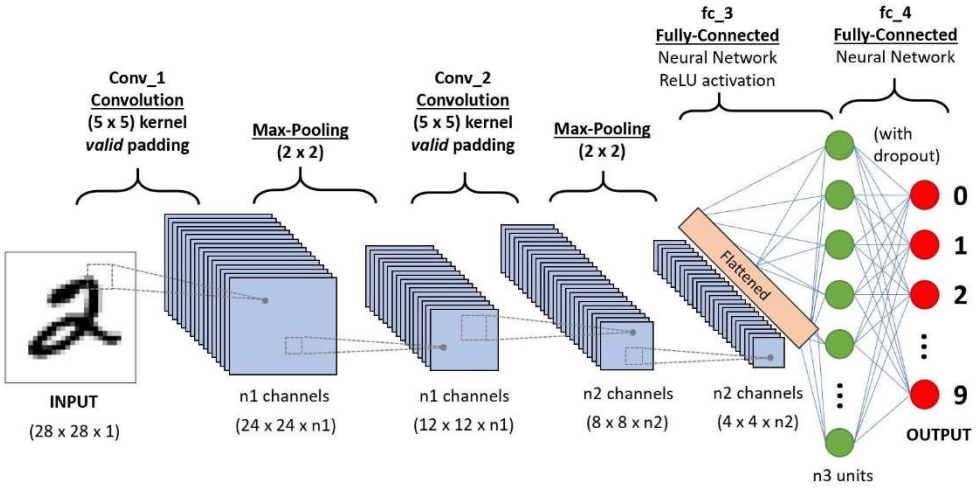

Figure 61. Typical covolutional layer with sequence to classify handwritten digits.

Though the layers are colloquially referred to as convolutions, this is only by convention (Shindel et al., 2019). Mathematically, it is technically a sliding dot product or cross-correlation. This has significance for the indices in the matrix, in that it affects how weight is determined at a specific index point (Yang et al., 2018).

The input is a tensor with shape (number of images) x (image width) x (image height) x (image depth). Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map width) x (feature map height) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- convolutional kernels defined by the width and height (hyper-parameters)

- the number of input channels and output channels (hyper-parameter)
- the depth of the convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (the opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradient problem in training traditional multi-layer neural networks with many layers using backpropagation.

4.5.2. POOLING

Similar to the convolutional layer, the pooling layer is responsible for reducing the spatial size of the convolved feature. This decreases the computational power required to process the data by reducing the dimensionality. Furthermore, it is useful for extracting dominant features that are rotational and positionally invariant, thus maintaining the effective training of the model. There are two types of pooling: max pooling and average pooling. Max pooling returns the maximum value from the portion of the image covered by the kernel. On the other hand, average pooling returns the average of all the values from the portion of the image covered by the kernel. Max pooling also acts as a noise suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, average pooling simply performs dimensionality

reduction as a noise suppressing mechanism. Hence, max pooling performs a lot better than average pooling as shown in Figure 62.

The convolutional layer and the pooling layer together form the *i*-th layer of a CNN. Depending on the complexities in the images, the number of such layers may be increased to capture more low-level details, but at the cost of more computational resources.

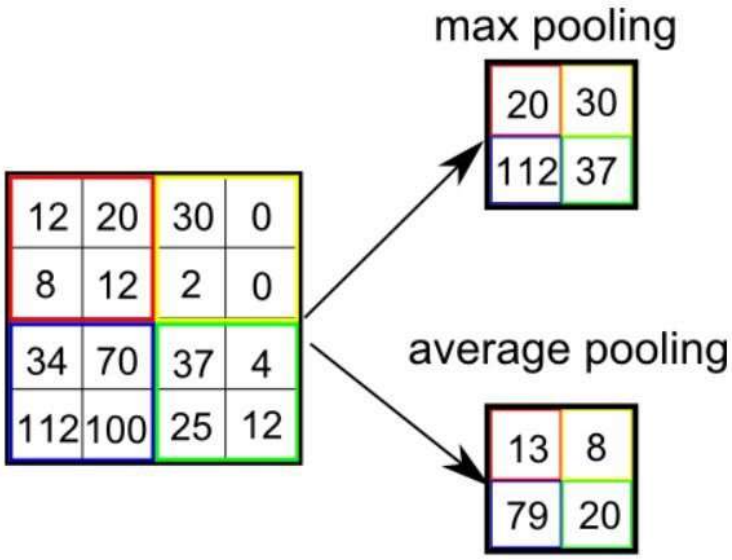

Figure 62. Types of pooling.

4.5.3. FULLY CONNECTED LAYER

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular neural networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset. Fully connected layers are an essential component of CNNs, which have proven very successful in recognizing and classifying images for computer vision. The CNN process begins with convolution and pooling, breaking down the image into features, and analyzing them independently. The result of this process feeds into a fully connected neural network structure that drives the final classification decision. As part of the convolutional network, there is also a fully connected layer that takes the end result of the convolution/pooling process and reaches a classification decision.

Figure 61 presents the layers needed to process the image of a written digit, with the number of pixels processed in every stage. Larger and more complex images would require more convolutional/pooling layers.

4.5.4. DESIGN OF THE COMPONENTS OF THE CNN-LSTM MODEL

The environmental prediction model developed in this study used data from sensors commonly used in greenhouses to estimate the electrical conductivity (EC) and volumetric water content (VWC) of the substrate and the evapotranspiration of the crop. The data was separated into a learning dataset and a validation dataset at a ratio of about 7:3 using data from 60 days from February to March 2019 and the same data from June to July of the same year. The test set was selected using data from three days per month.

The basic structure of the environmental model proposed in this study is a deep-learning model combining a CNN with LSTM. Figure 63 presents the overall structure of the model. As shown in Figure 63(a), two input datasets were used in the model, The first was used to predict the EC and VWC of the substrate based on internal and external greenhouse conditions and the nutrient irrigation control history. The second dataset contained crop condition and drainage data and was used to estimate evapotranspiration in conjunction with the input variables from the first dataset.

The input data took the form of a two-dimensional (2D) array using about 30 consecutive data points. A total of four synthetic layers were used for the 2D convolutional layers and a kernel size of (1, 3) was used. A kernel size of (1, 2) was used for each layer, and max pooling was used for pooling between composite layers. The pool size for each layer was (1,3), (2,1), and (1,2), and the ReLU activity function was used. The number of nodes in the dense layer was 15, and ReLU was used as the active function. The model was trained with a combination of 100 epochs and 5 batch sizes. The optimization function was set to the default value of Adadelta. A dropout rate of 25% was set for each floor to contribute to the learning rate.

LSTM architecture can eliminate memory loss in continuous time-series data. In this learning model, the LSTM architecture in the latter part of the model inferred from the 2D CNN is combined.

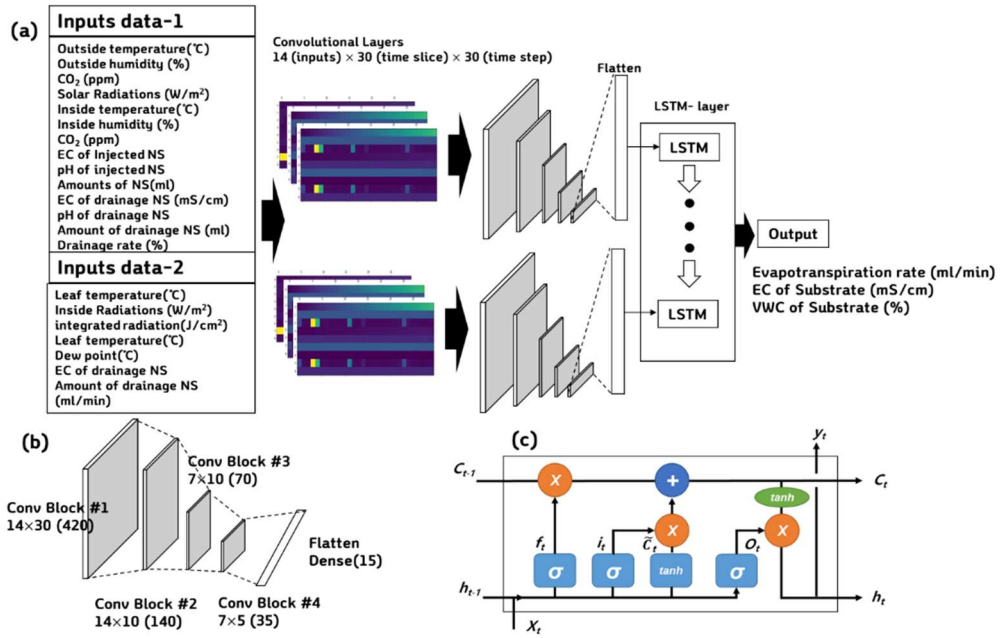

Figure 63. Overall conceptual diagram of CNN-LSTM based environmental prediction model (a), 2 D CNN conv layer diagram (b), LSTM device (c).

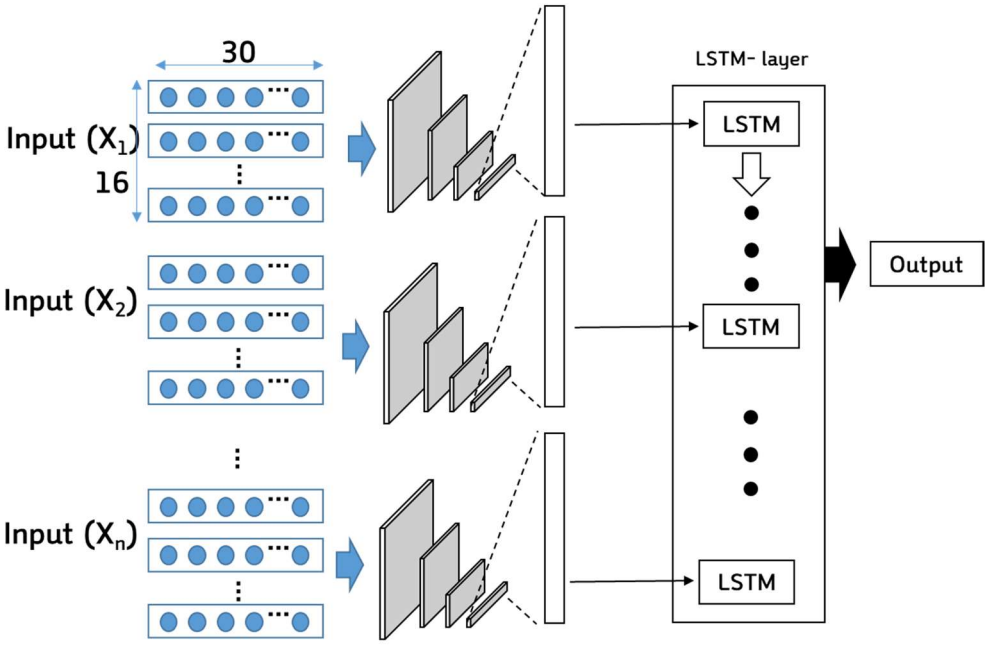

Figure 64 The details of CNN-LSTM structure.

The proposed CNN-LSTM model was developed using TensorFlow and Keras in Python 3.7 and compared with the performance of the RNN-LSTM model. The comparison model had the same input, output data, and time step size, two hidden layers with 30 and 20 nodes, and an ReLu activity function.

4.5.5. THE RESULTS OF CNN-LSTM BASED CLIMATE MODELS

In the present study, ANN, RNN-LSTM, and CNN-LSTM models were developed to predict environmental changes in the greenhouse, and their performance was compared. Although the RNN-LSTM model proposed in the time-series analysis performed better than the basic neural network structure, the best-performing approach was the CNN-LSTM model. This is because of the optimal combination of variables in the CNN model and the use of dropout and batch normalization techniques to avoid overfitting. Combined with the LSTM architecture, it outperformed the conventional RNN-LSTM model. The results for temperature, CO_2, and humidity can be seen in Figure 65, Figure 66, and Figure 67, respectively. Statistical comparisons for the results are summarized in Table 24.

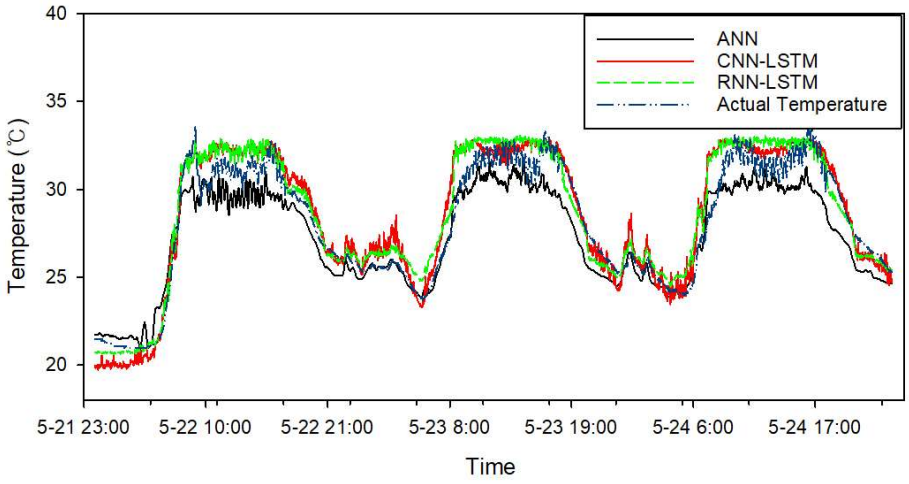

Figure 65. Comparison of the predictive models for the change in temperature after 30 minutes in the greenhouse.

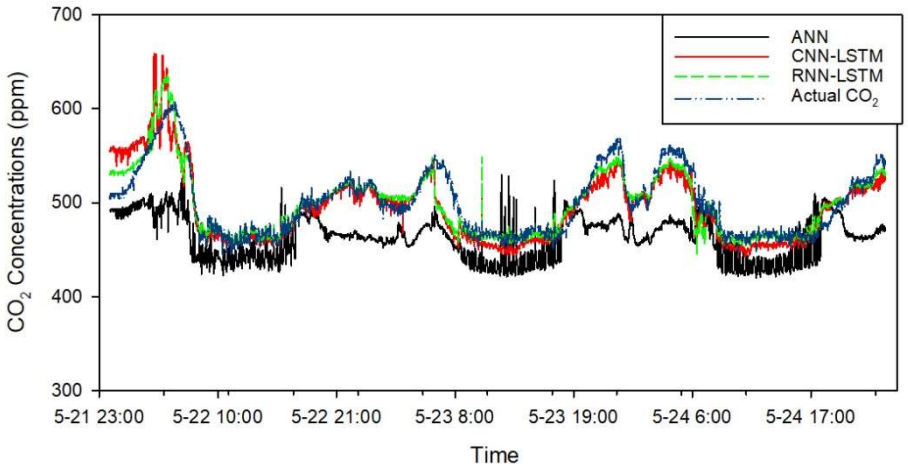

Figure 66. Comparison of the predictive models for the change in CO_2 after 30 minutes in the greenhouse.

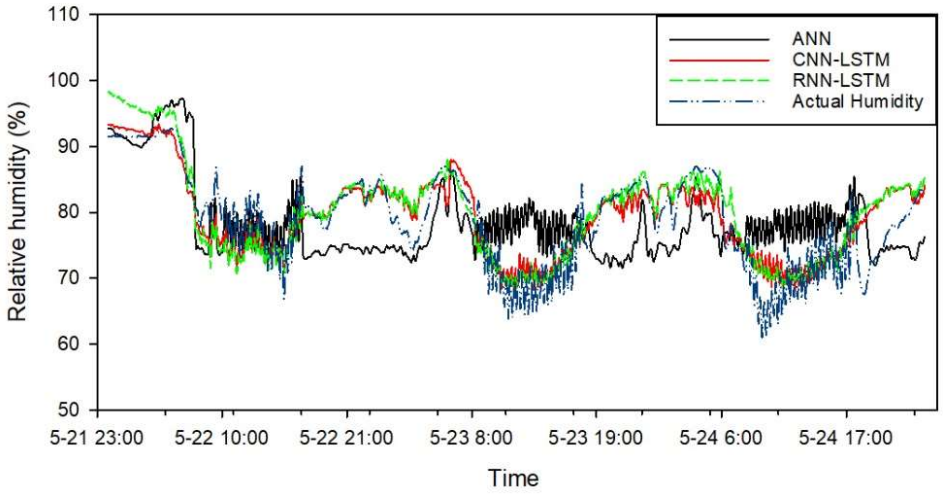

Figure 67. Comparison of the predictive models for the change in relative humidity after 30 minutes in the greenhouse.

Table 24. Performance comparison for the predictive models for the change in climatic conditions after 30 minutes in the greenhouse

	ANN			CNN-LSTM			RNN-LSTM		
	RMSE	SEP (%)	R^2	RMSE	SEP (%)	R^2	RMSE	SEP (%)	R^2
Temperature	1.09	3.99	0.92	0.83	3.10	0.94	0.82	3.03	0.95
Humidity	8.52	11.27	0.54	5.79	8.56	0.82	5.86	8.91	0.84
CO_2	39.94	12.78	0.45	17.37	5.55	0.85	18.86	6.01	0.79

4.5.6. THE RESULTS OF CNN-LSTM BASED SUBSTRATE MODELS

Prediction of the environmental conditions in the substrate has not been carefully considered in conventional watering control. In this study, the CNN model was developed to predict not only the climatic conditions but also the EC and VWC of the hydroponic substrate in the target greenhouse. The performance of the CNN-LSTM model and the RNN-LSTM models in the prediction of the EC (Figure 68) and VWC (Figure 69) was compared, with the CNN-LSTM model exhibiting the best performance, as summarized in Table.25.

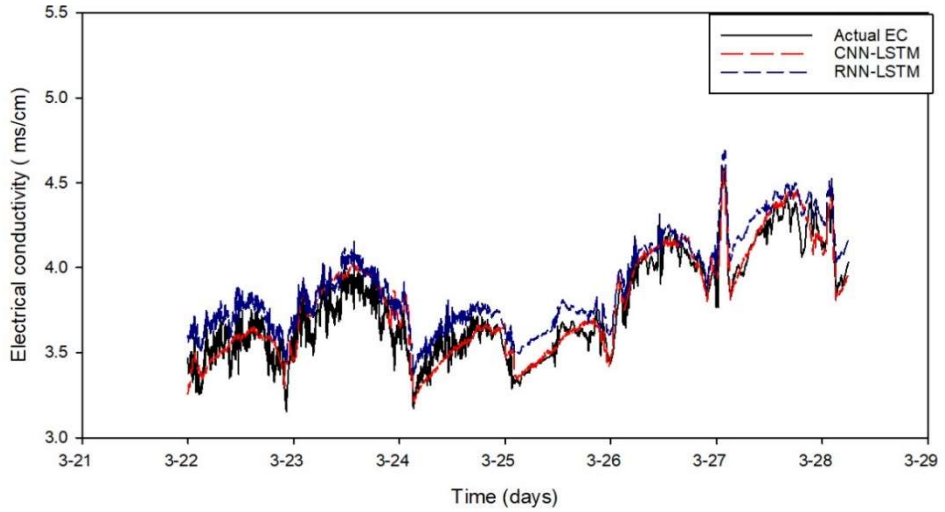

Figure 68. Comparison of the prediction models for the electrical conductivity (EC) of the hydroponic growing substrate.

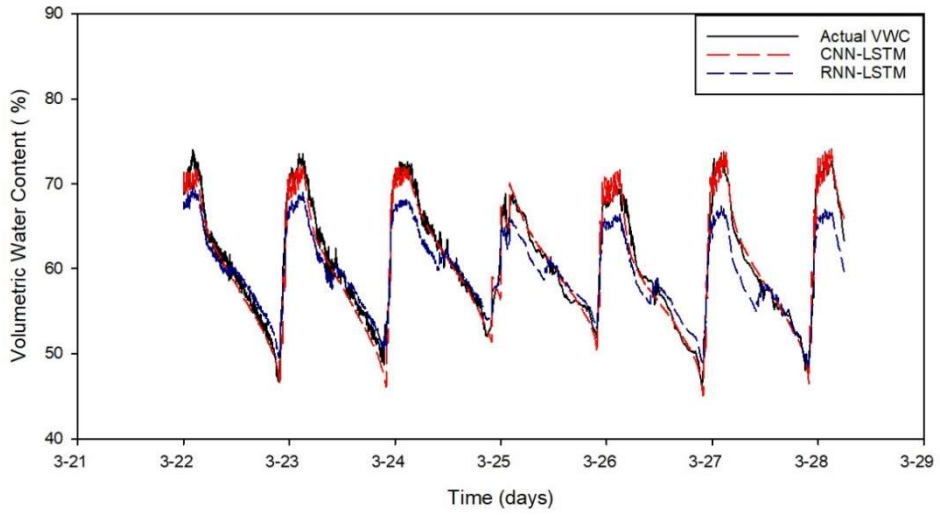

Figure 69. Comparison of the prediction models for the volumetric water content (VWC) of the hydroponic growing substrate.

Table 25. Summary of the performance of the predictive models for the change in the EC and VWC of the hydroponic substrate after 30 minutes in the greenhouse.

	CNN-LSTM			RNN-LSTM		
	RMSE	SEP (%)	R^2	RMSE	SEP (%)	R^2
EC	0.07	1.89	0.94	0.13	3.60	0.92
VWC	1.01	1.66	0.95	2.10	3.45	0.89

The evapotranspiration rate of the greenhouse tomato crop was calculated using measurements taken by a device that monitored the weight of the hydroponic medium. Based on the change in the weight of substrate, water supply, and drainage, the evapotranspiration rate was calculated as shown in Figure 70. The evapotranspiration component of the CNN-LSTM model was developed based on data from 60 days. The actual evapotranspiration rate and that predicted by the model were compared; the proposed CNN-LTSM model demonstrated excellent predictive

performance with an RMSE of 0.14 ml/min (Figure 71).

.

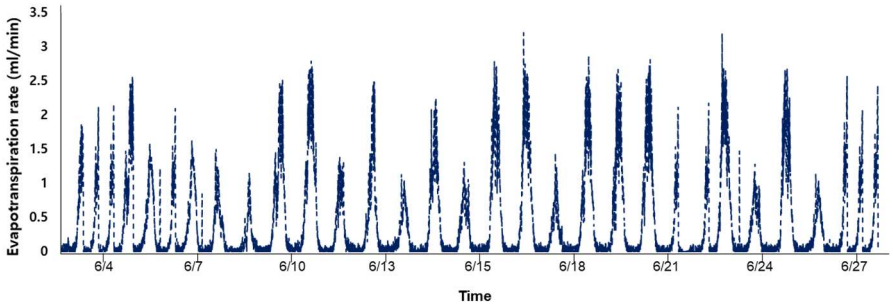

Figure 70. Evapotranspiration monitoring dataset based on the weight sensor for the growing medium used to train the CNN-LSTM model.

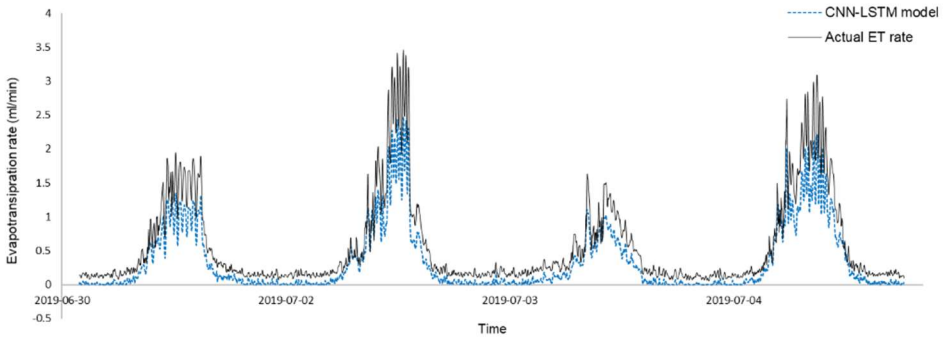

Figure 71. Comparison of the predictive results for the proposed CNN-LSTM model and the actual evapotranspiration rate.

4.5.7. COMPARISON OF TRAINING PERFORMANCE BETWEEN RNN-LSTM AND CNN-LSTM

Mathematically, convolution is a grouping formula. In a CNN, convolution occurs between two matrices to deliver a third output matrix. The matrix is simply a rectangular array of numbers stored in columns and rows. A CNN utilizes the convolution in the convolutional layers to segregate the input information and identify the actual information (Pandey and Janghel, 2019). The convolutional layer in a CNN is engaged in highly complicated computational activity, acting as a

numerical filter that helps the computer to identify the corners of an image, concentrated and faded areas, color contractions, and other attributes such as the height, size, and weight of the image, the depth, and pixel scattering. In a CNN, a pooling layer is often placed between the convolution layers to reduce the size of the representations from the convolutional layers, thus decreasing the memory requirements and allowing the use of many convolutional layers.

CNNs contain numerous layers of filters or neuron layers that are hidden and optimized, increasing their efficiency in detecting an image. Because of this popular feature, they are called a feedforward loop (Fan et al., 2016). CNNs are specifically designed for computer vision tasks, but training them with appropriate data can also allow them to correctly classify music, speech, videos, and text (Kim and Cho, 2019).

RNNs have the same traditional structure as ANNs and CNNs. However, they have another memory partition that can act as a feedback loop. Similarly, like the human brain, especially in conversations, high weight is given to the redundancy of data to relate and understand the sentences and the meaning behind them. This unique feature of an RNN is used to predict the next set or sequence of words. An RNN can also be fed sequences of data of varying lengths and sizes, while a CNN operates only with fixed input data (Zhang and Wang, 2001).

The prediction performance of the RNN-LSTM and CNN-LSTM models in the previously developed greenhouse environment prediction model was compared, and the two models produced equivalent results. Therefore, when these two models are used in field applications, it is necessary to compare their training performance, including the time required for training.

The daily greenhouse data used for training contained a total of 16 input variables (Table 26). There were 14,400 data points in total arranged in 10 mini-batches for each model. The output of the model was the temperature inside the greenhouse, and the performance of each of the two models was compared. A desktop computer (Intel® core ™ i9-9900K CPU @ 3.60 Hz, NVIDIA GeForce RTX 2080 Ti) and a Jetson Nano embedded board (NVIDIA Maxwell™ architecture with 128 NVIDIA CUDA® cores 0.5 TFLOPs, Quad-core ARM® Cortex®-A57 MPCore processor) were compared for training time.

Table 26. Ranges for input data used for building a prediction model of temperature.

Input variables (unit)	Range
Outside temperature (℃)	18.5−29.9
Outside humidity (%)	38.5−56.5
Outside CO_2 concentration (ppm)	371.4−402.0
Radiation (W/m^2)	0−1155.4
Wind speed (m/s)	0−0.31
Wind direction (°)	0−359
Fan (0, 1)	0, 1
Heating (%)	0
Fogging (0, 1)	0, 1
CO_2 injection (0, 1)	0, 1
Heat retention curtain (%)	80−100
Shade curtain (%)	0−100
Left window (%)	0−100
Right window (%)	0−100
Inside humidity (%)	33.2−67.5
Inside CO_2 concentration (ppm)	325.5−501.2

Table 27. Training times for the CNN-LSTM and RNN-LSTM models.

	CNN-LSTM		RNN-LSTM	
	Training time (seconds)	**R^2**	**Training time (seconds)**	**R^2**
Jetson Nano	251.0	0.99	145.2	0.99
Desktop with GPU	45.5	0.99	33.4	0.99
Desktop CPU only	242.1	0.99	222.1	0.99

Table 27 presents the training time required for 1440 data points using the CNN-LSTM and RNN-LSTM models. Overall, the RNN-LSTM model was faster than the CNN-LSTM model. With a GPU-equipped desktop, the CNN-LSTM model required 45.5 seconds, while the RNN-LSTM model required only 33.4 seconds. Using the Jetson Nano board, 251.0 and 145.2 seconds were required, respectively. However, in the CNN-LSTM structure, there is room for reducing the training time

further by adjusting the dropout, mini-batch, and kernel settings in the convolutional layers.

In addition, the CNN structure has several advantages over general neural networks. For example, a CNN can express the connectivity between the input data well and can evaluate the importance of data in the early stages. Li and Liu (2019) employed the information bottleneck (IB) theory to understand the dynamic behavior of CNNs and investigated how their fundamental features impacted their performance. In a series of experiments using the MNIST database, they demonstrated that the compression phase is not observed in all cases. This means that CNNs exhibit more complicated behavior than do feedforward neural networks. In addition, it is likely that future climate prediction models for greenhouses will need to be able to handle image data from crops. As a starting point for this, a deep neural network (DNN) model based on the CNN architecture will be a possible option.

4.5.8. STUDY TO IMPROVE THE HUMIDITY PREDICTION PERFORMANCE CONSIDERING THE AMOUNT OF EVAPOTRANSPIRATION RATE (ET)

Accurately predicting the humidity within a greenhouse is more challenging than predicting the temperature (He and Ma, 2010). Because the evapotranspiration rate for a crop is associated with the humidity of the greenhouse (Orgaz et al., 2005; Stanghellini, 1992; Federico Villarreal-Guerrero et al., 2012), this study intended to improve humidity prediction performance by adding information about evapotranspiration to a data-based CNN model. Various studies have been conducted on the prediction of the evapotranspiration of crops in greenhouses because it is an important variable in irrigation strategy and crop management (Shin et al., 2014).

Traditionally, many studies have been based on the Penman-Monteith equation (FAO 56-PM), which is the single standard method recommended by the FAO for the computation of ET from complete meteorological data (Beven, 1979; Chiew et al., 1995). Stanghellini (1987) revised the Penman-Monteith evapotranspiration model to represent the conditions in a greenhouse, in which air velocities are typically less than 1.0 m/s. The model also includes more complex equations to calculate internal and external resistance. To apply the model in a greenhouse environment, Stanghellini (1987) focused on the energy exchange from the leaf and expanded it to the entire plant. The radiation absorption by multi-layered canopies is also considered by applying the leaf area index (LAI). Villarreal-Guerrero et al. (2012) reported that the Stanghellini model produced the smallest deviation between the predicted and measured evapotranspiration in a greenhouse for tomatoes.

The present study aimed to improve the prediction of humidity in a greenhouse by adding Stanghellini's evapotranspiration model to the CNN-LSTM model.

$$LE \cong \frac{2 \cdot LAI \cdot \rho_a c_p}{\gamma \cdot (r_i + r_e)} (VPD) \quad \dots [37]$$

Here, Eq. [37] can be converted to

$$LE \cong \frac{2 \cdot LAI \cdot \rho_a c_p}{1 + \frac{\delta}{\gamma} + \frac{r_i}{r_e}} \left[0.07 \frac{\delta}{\gamma} \frac{I_s}{\rho_a c_p} + 0.16 \frac{\delta}{\gamma} \frac{T_h - T_o}{r_R} + \frac{1}{r_e} \frac{e_a^* - e_a}{\gamma} \right] \quad \dots [38]$$

Table 28. List of variables used to determine the evapotranspiration rate using the Stanghellini model.

Symbol	Variable	Unit
E	Evapotranspiration rate	kg sec^{-1} m^{-2}canopy area
T_{air}	Ambient air temperature	℃
T_o	Temperature at the leaf surface	℃
RH	Relative humidity	%
I_s	Shortwave irradiance	W m^{-2}
LAI	Leaf area index; the ratio of the total leaf area (one side) to the canopy area, 2.5 in this study	m^2 m^{-2}
L	Latent heat of the vaporization of water, 2,502,535.239–2385.76 · T_{air}	J· kg^{-1}
ρ_a	Air density, 100,000/ 287 ·(T_{air}+273.16)	kg m^{-3}
c_p	Air specific heat at constant pressure, 1013	Jkg^{-1} ℃$^{-1}$
δ	Slope of the saturation vapor pressure–temperature curve, 41.45 exp(0.06088· T_{air})	Pa ℃$^{-1}$
γ	Psychometric constant, $\frac{c_p}{L}\ \frac{P_{atm}}{0.6216}$	Pa ℃$^{-1}$
P_{atm}	Atmospheric pressure, 101,325 $(\frac{293-0.0065\ \cdot h}{293})^{5.26}$	Pa
h	Elevation above sea level, 70 m	
r_i	Internal resistance of the canopy to vapor transfer $r_i = \frac{I_s + 4.30}{I_s + 0.54}[1 + 2.3 \cdot 10^{-2}(T_0 - 24.5)^2] \cdot \tilde{r}_i(CO_2) \cdot \tilde{r}_i(e_0 - e_a)$	S m^{-1}
$\tilde{r}_i(CO_2)$	$= 1,\ I_s = 0$ $= 1 + 6.1 \cdot 10^{-7}(CO_2 - 200)^2,\ CO_2 < 1$ $= 1,\ CO_2 \geq 11$	

	$\tilde{r}_i(e_0 - e_a) = 1 + 4.3 \cdot (e_0 - e_a)^2, e_0 - e_a < 0.8\ kPa$ $= 3.8, \quad e_0 - e_a \geq 0.8\ kPa$ External resistance of the canopy to sensible heat transfer,	
r_e	l =15 cm (5–20 cm) $r_e = \frac{1174\ l^{0.5}}{(l\lvert T_0 - T_a\rvert + 207\ u^2)^{0.25}}$	
T_h	Apparent temperature of the ambient environment as determined by the pipe, floor, and cladding temperature, T_{air}	
r_R	Linearization factor of the radiation heat flux equation, $\frac{\rho_a c_p}{4 \cdot \sigma \cdot (T_a + 273.15)^3}\ [s\ m^{-1}]$	
σ	Stefan-Boltzmann constant, $5.669 \cdot 10^{-8}$	J K^{-4} m^{-2} s^{-1}
e_a^*	Saturation vapor pressure at mean air temperature, $610.78 \cdot exp\left(\frac{17.269 \cdot T_{air}}{237.3 + T_{air}}\right)\ [Pa]$	Pa
e_a	Vapor pressure at air temperature, $e_a^* \frac{RH}{100}\ [Pa]$	Pa

Three versions of the CNN model were trained using data from 21 days from June 1 to June 21 and verified using test data from 2.5 days. Table 29 presents the datasets used in the three CNN models. Figure 73 shows the trend in the evapotranspiration rate, which was used in Model 2, and the trends in the additional sensing information used in Model 3 are presented in Figure 74.

Table 29. The datasets used for the humidity prediction model

Model	Data
Model 1 (previously developed set)	Inside and outside climate sensors, all actuator signals
Model 2 (+ evapotranspiration)	Inside and outside climate sensors, all actuator signals, evapotranspiration rate
Model 3 (+ root-zone and leaf temperature sensor)	Inside and outside climate sensors, all actuator signals, leaf surface temperature, substrate weight, dew points

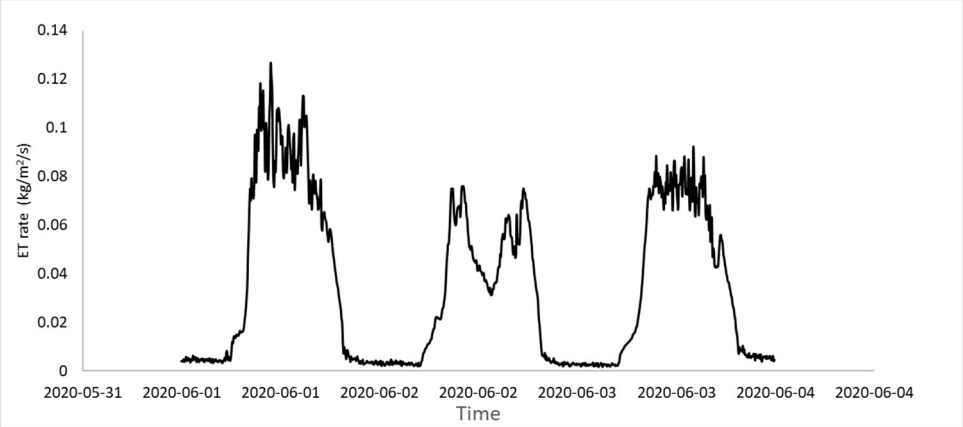

Figure 72. Evapotranspiration rate calculated using the Stenghilini evapotranspiration model (used in Model 2).

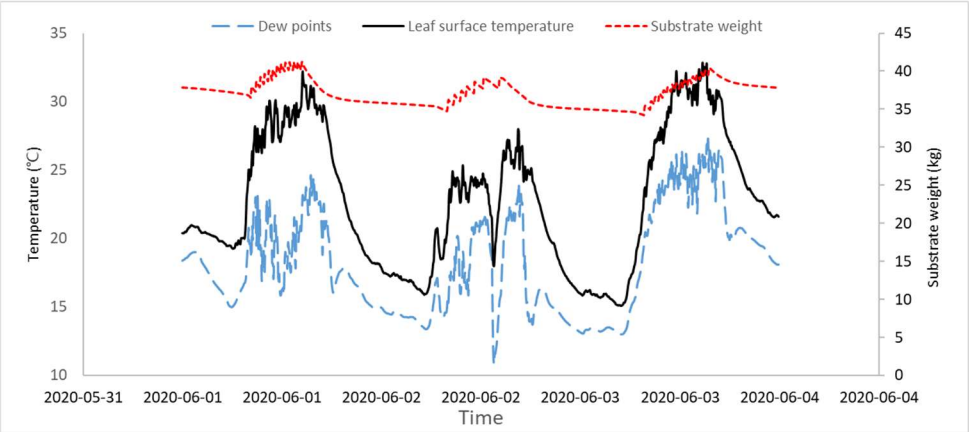

Figure 73. Dew point temperature, leaf surface temperature, and substrate weight (used in Model 3).

In the training stage, all three models produced an R^2 of more than 0.95 (Figure 74), with the highest observed for Model 3 (0.98). In the test set, Model 2, which included the evapotranspiration rate, demonstrated the best prediction performance with an RMSE of 3.16. Overall, the standard error of prediction (SEP) improved from 9.12% in Model 1 to 5.78% and 6.12% for Models 2 and 3, respectively (Figure 75). This means that the relative humidity and the evapotranspiration rate in the greenhouse are closely related to each other, and the more information related to moisture that can be collected from the greenhouse, the more accurate humidity prediction will be. In the future, a hybrid of a deep-learning model based on data and a physical model should be investigated.

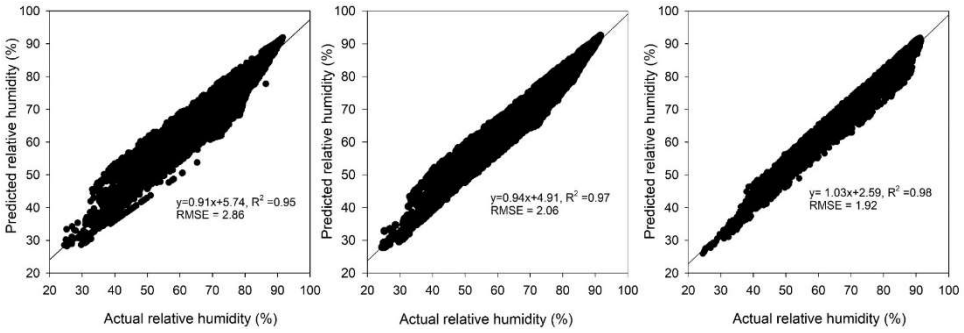

Figure 74. Training results for the three models (left: Model 1; middle: Model 2; right: Model 3).

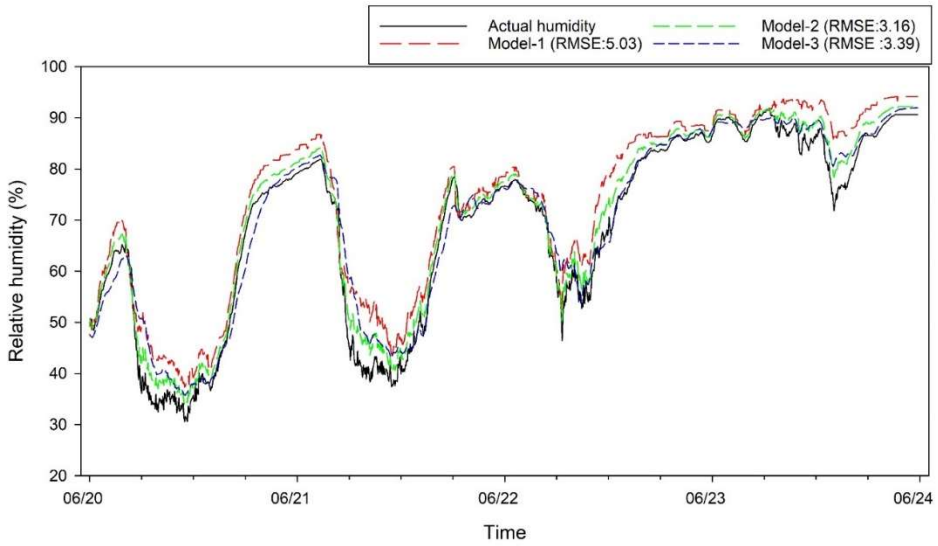

Figure 75. Test set results for the three models in predicting the relative humidity in a greenhouse.

4.6. AUTOMATED LEARNING SYSTEM FOR DEEP LEARNING MODELS BASED ON EMBEDDED BOARD

Greenhouses artificially control the environment to provide optimal conditions for crop growth and fruit development. In addition, by actively responding to various external climate changes, it is possible to stably cultivate crops by creating an internal environment, which can play a very important role in securing future food resources. The internal environment of the greenhouse includes temperature, humidity, light quantity, and carbon dioxide concentration. For this reaseon, strategic inside climate management needs to be carried out to promote photosynthesis, evapotranspiration, crop growth and fruit development.

The climatic variables inside a greenhouse are closely related to each other, so it is not easy to interpret them individually. Therefore, although many studies have investigated environmental change modeling based on energy conservation laws (Norton et al., 2007), few of these techniques have been applied in the field and used directly in greenhouse management. For this reason, it is difficult to input exact numerical values for physical conditions such as the structure and construction

material for the greenhouse. To apply the results for one specific greenhouse to other greenhouses, accurate model analysis can only be conducted by entering the initial setting values. In addition, active model inference is limited by the wide variety of possible environmental changes (Bartzanas et al., 2002).

Recently, with the development of ICT technology, many greenhouses have installed various sensors, meaning that data on the internal environment and crops can be acquired in real-time and used for big-data analysis. These changes are based on sensor data for the greenhouse environment management approach, and control algorithms based on measured environmental variables have been developed. In addition, methods for inferring the evapotranspiration rate of crops, which is otherwise difficult to measure, and for modeling growth have been proposed using data-based environmental modeling. Data-based modeling has been facilitated by the recent development of deep-learning technology and computing power, and it has been employed in various industries, particularly in agriculture. Indeed, they have been employed to model nonlinear environmental changes in greenhouses and have been reported to perform well.

Recently, RNN models have been widely used, with these time series neural network models employed not only for greenhouse environmental changes but also for the LAI and evapotranspiration (Anapalli et al., 2016; Libardi et al., 2019). In addition, RNN-LSTM and NARX models have been proposed to predict environmental changes 30 minutes in the future inside a greenhouse, exhibiting better performance than previous models by utilizing ventilation control based on the predicted temperature. Various deep-learning models using CNNs have also been developed and applied outside of greenhouses (Hongkang et al., 2018). CNNs are particularly useful in image recognition. CNNs can also be applied to regression with excellent outcomes (Grinblat et al., 2016). For this reason, it is possible to develop a model for the prediction of the internal climate of greenhouses based on a CNN-LSTM model.

4.6.1. EXPERIMENTAL ENVIRONMENT AND CONTROL CONDITIONS

The experiment was conducted during March, which is the most important period for temperature control in Korea and a period when both heating and ventilation are required to maintain the temperature. Temperature control in the greenhouse cultivation environment was managed. Target temperatures were set for each of the six time periods (P_1 to P_6), and ventilation control was configured accordingly. The humidity was supplemented with fogging at a level of at least 40% starting from P_3 to P_5. Sensor modules used to measure the internal temperature, humidity, and CO_2 levels in the greenhouse were installed and environmental data were obtained every minute, while a weather station was installed to monitor the external temperature, humidity, solar radiation, wind direction, wind speed, and rain. The environmental control signals included the left windows, right windows, the heat retention curtains, and the shading curtains, which were all operated on a proportional basis. In contrast, the fans, foggers, CO_2 injectors, and electric heating had simple on/off valve control. The environmental data and control signals were connected, with a Raspberry Pi 3 computer performing the central calculations to determine the monitoring and control signals. The monitoring environment was based on the open-platform FarmOS. The signal obtained from the sensor was first converted to an ADC or digital signal and then stored in the database.

4.6.2. DEEP-LEARNING MODEL IMPLEMENTATION AND AUTOMATIC LEARNING ALGORITHM

The goal of this experiment was to develop a model that could predict climatic changes in the greenhouse and for the output of this model to be used in adjusting the greenhouse settings. In total, 122,400 data points were obtained over 85 days from January to March 2020 for the training process. The CNN and RNN models were tested on three consecutive days of data from March 12 to 15 to compare their accuracy.

The CNN-LSTM and RNN-LSTM models tested in this experiment had already been developed based on three months of training data, but experiments are needed to determine the accuracy change in all environments and changing sites. Therefore, in this study, model fitting was conducted daily in the field using the two model structures that were trained offline and showed the best prediction performance (Figure 76). The models were updated daily, and the model weights were tuned using 1,440 data points from the previous day.

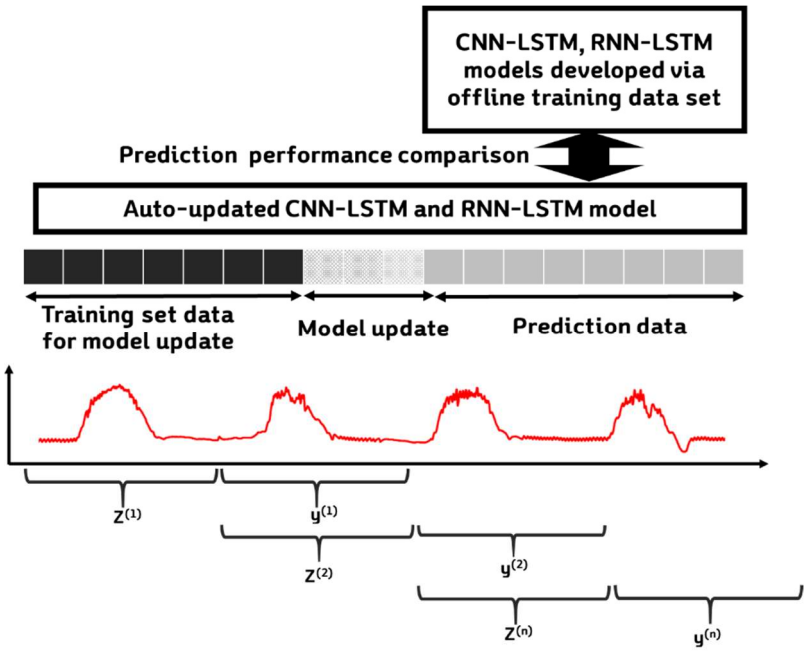

Figure 76. Sampling from the original time-series for climate prediction models

As shown in Figure 76, smaller sub-samples from the original time series were defined, and a moving window was used to sample from this series. This can be extended to a multi-variate case. Here, T is the sequence length, q is a positive integer that indicates the number of steps ahead to be predicted, and N is the total number of subsamples, which depends on the length of the original time series and T. Figure 77 illustrates how LSTM models process sequential data. The values for the gates, memory cells, and hidden states are updated recurrently.

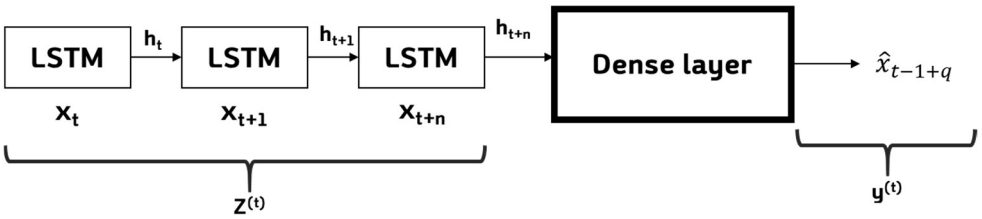

Figure 77 Unfolded LSTM layer.

Training of the model was conducted between 12:00 AM (midnight) and 1:00 AM, and the accuracy of the fitted model was tested by predicting the environmental changes that day. During training, the environmental and control data were continuously predicted, but environmental change prediction through model inference was paused during the updating process. A flow chart for the learning process is presented in Figure 78. A total of four models were compared in terms of their accuracy: the automatically trained RNN-LSTM and CNN-LSTM models, and the same two models trained offline (Figure 78).

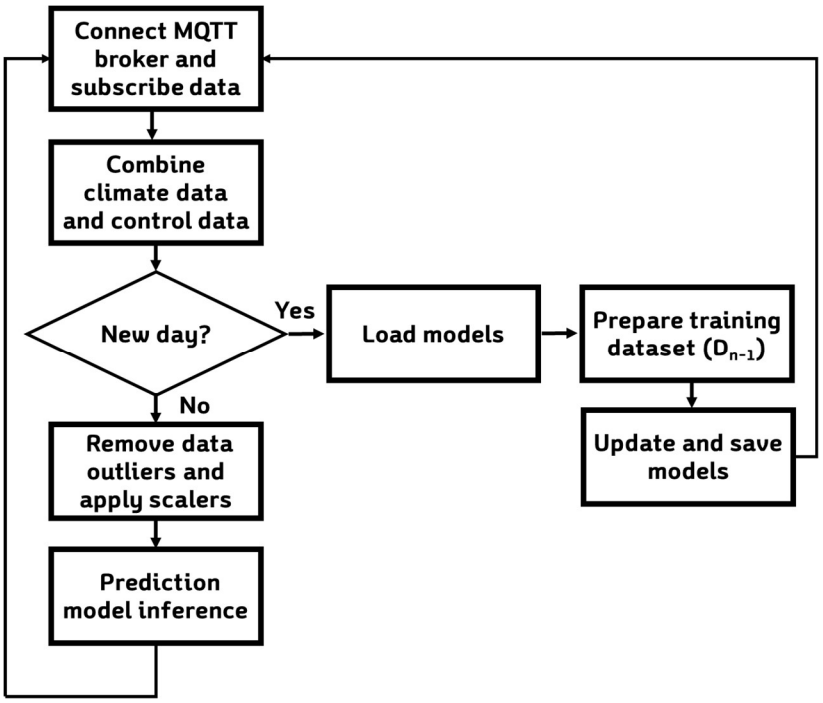

Figure 78. Flow chart for the automatic model training system.

4.6.3. THE RESULTS OF AUTOMATED LEARNING SYSTEM

The IoT-based automatic model learning system was implemented based on actual greenhouse data for 14 days. All of the sensor and actuator information was updated and stored every minute, with a total of 1440 data points per day used to update the model. The results were compared with the results for the models trained offline, and the prediction accuracy was compared based on the SEP (%). The online model was used for model inference by substituting the current situation values with the updated model at 1:00 AM on the day of the temperature measurement.

Figure 79 shows the results after three days of training for the temperatures predicted by the offline and online models. The solid red line shows the actual temperature. On March 8, there were no results for the online models because they were trained based on data from Day 1 to Day 2. The RMSE on Day 4 for the offline RNN and CNN models was 1.43 and 1.55, respectively, while that for the online models was 4.64 and 5.89. There was a significant deviation for the online models on Day 2, but it was difficult to predict the temperature changes on that day in the model using the training set created on Day 1. This was because it rained on the same day, and it was confirmed that external humidity and rainfall were not reflected in the temperature predictions.

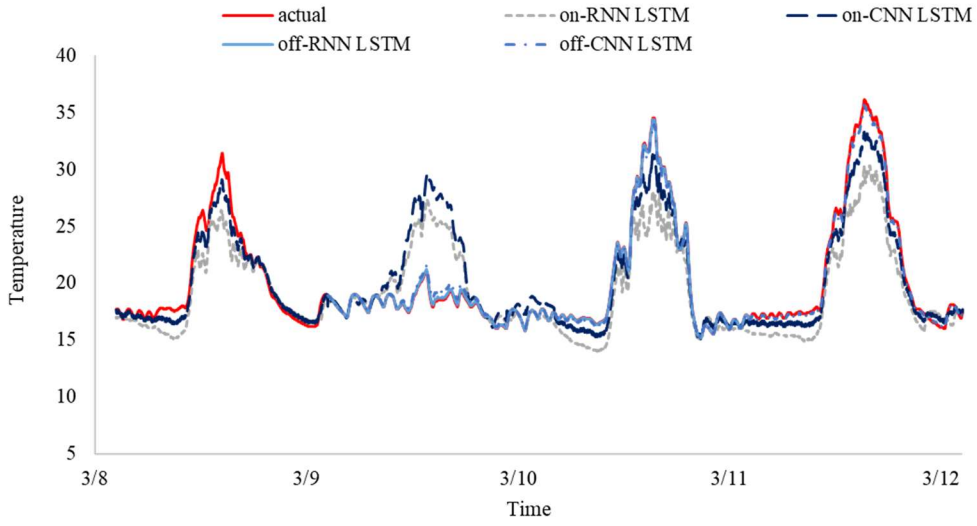

Figure 79. Temperature prediction curves of four deep learning models with actual values at 1-4 days.

The forecasting performance of the four models from March 12 to 15 is presented in Figure 80. The RMSEP for temperature during this period was 1.54 for both of the offline RNN and CNN models and 1.95 and 2.33 for the online models, respectively.

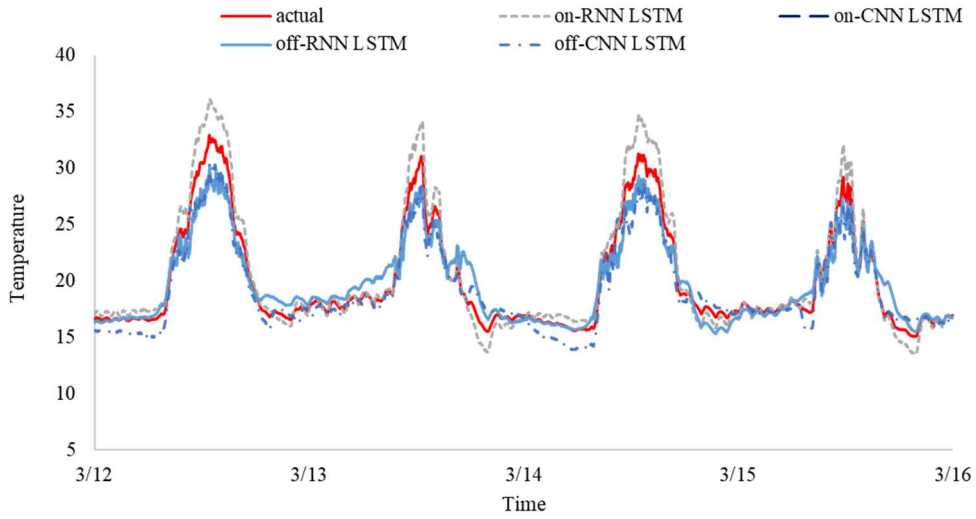

Figure 80. Temperature prediction curves for the four deep-learning models compared with actual data from Days 5 to 9.

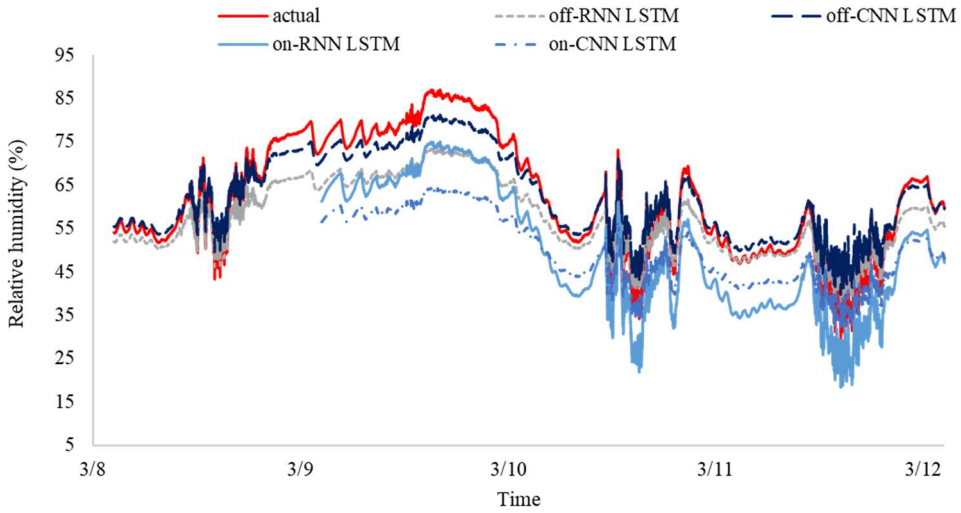

Figure 81. Relative humidity prediction curves for the four deep-learning models compared with actual data from Days 1 to 4.

Figure 81 presents the humidity predictions for the online and offline RNN and CNN models for the first four days. Overall, the accuracy of the humidity prediction

in terms of tracking the actual data was low for both the online and offline models. The offline RNN and CNN models had an RMSEP of 9.77% and 8.49%, respectively, while the online models had an RMSEP of 11.55% and 14.55%.

Figure 82 displays the humidity prediction performance from Days 5 to 9. Overall, although the prediction accuracy was still relatively low, the predictive performance tended to improve over time. The RMSEP was 5.23% and 6.39% for the offline RNN and CNN models, respectively, and 10.15% and 7.23% for the online models. The online models tended to be vulnerable to sudden increases in humidity, possibly due to the lack of training data for sudden rain or increases in external humidity.

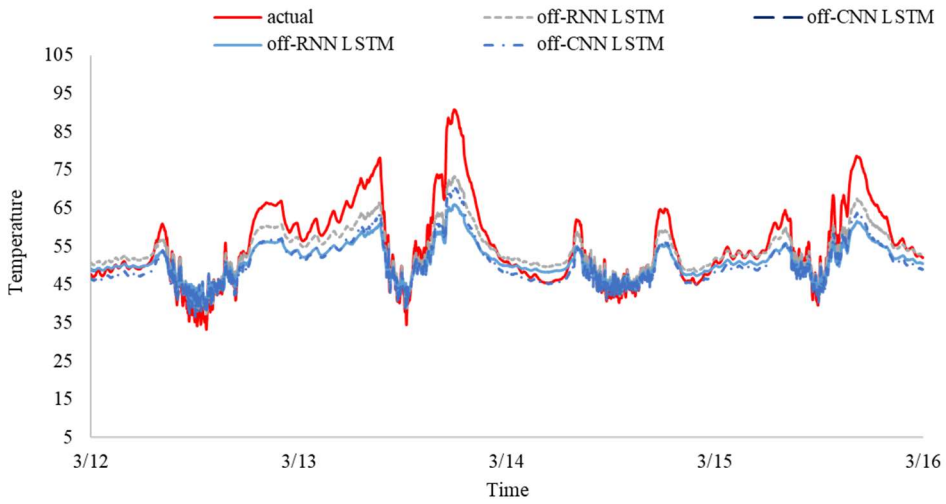

Figure 82. Relative humidity prediction curves for the four deep-learning models compared with actual data for Days 5 to 9.

Figure 83 compares the prediction performance for temperature and humidity the next day using the automatic learning models. It presents the R^2 values obtained by comparing the predicted and actual values for the four models each day. The temperature prediction performance of the online models was initially low, but it reached an R^2 of 0.85 or higher after four days, with the performance level becoming similar to that of the offline models after around 9 to 12 days. It was thus demonstrated that the prediction performance of the online automatic learning models was more accurate than that of the offline models. Although the prediction performance for humidity was lower than that for temperature as a whole, the accuracy of the online models tended to continuously increase over time. The

mechanisms underlying the humidity levels inside the greenhouse are more complex than those for temperature, and significant errors were observed due to the effect of external influences, which are difficult to predict.

To summarize, RNN-LSTM and CNN-LSTM models were employed to predict changes in the temperature and humidity of a greenhouse, which is important for controlling the internal climate. The models stored environmental and control history data collected from the greenhouse every minute and these data were used to tune the model at a specific time every night. The accuracy of these models increased over time as this process was repeated. After 9 to 12 days, the prediction performance of the online models was the same as that for the models that had been trained offline using 60 days of data. This proposed automatic learning model thus appears to be convenient for use in greenhouse management systems.

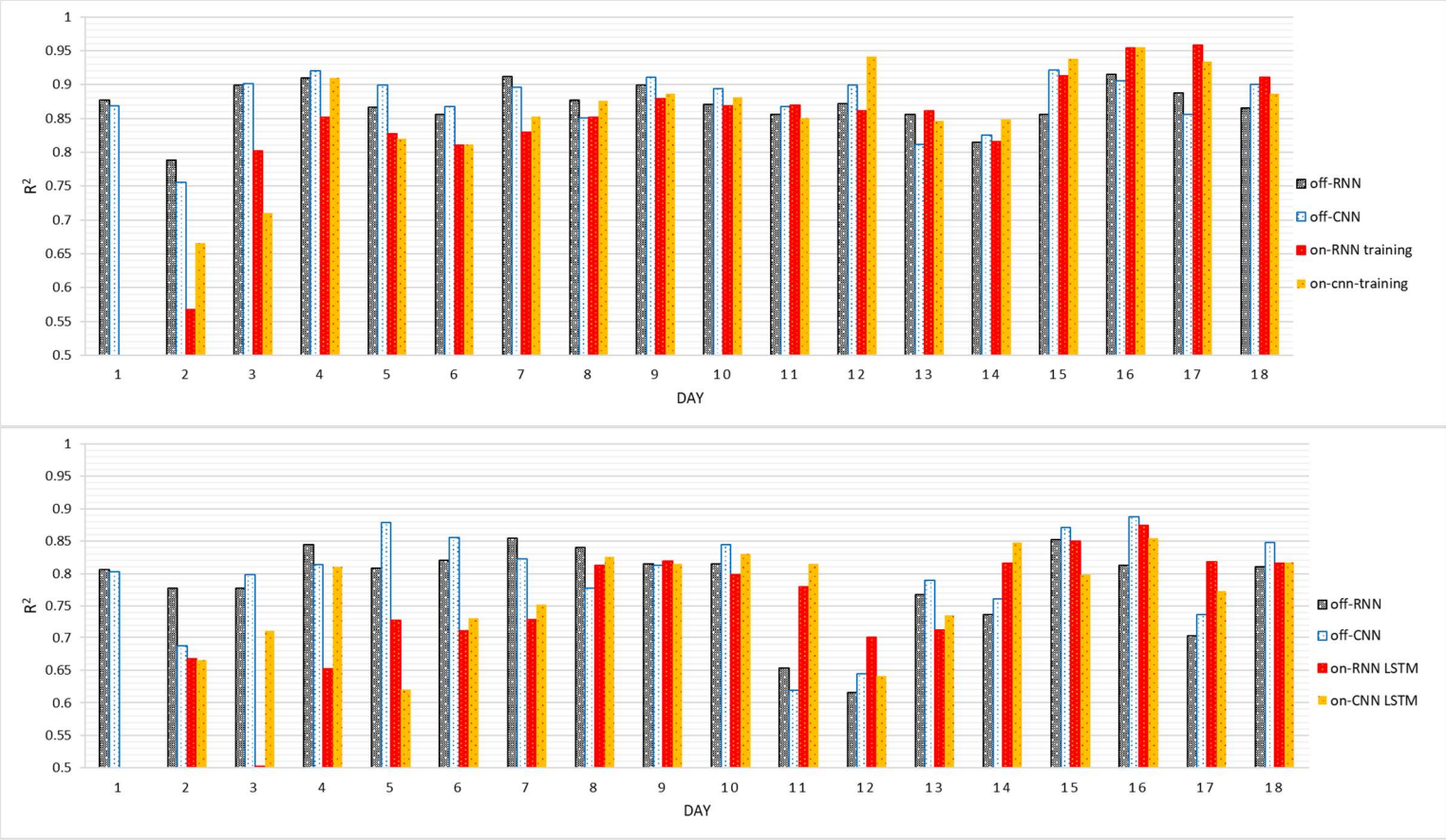

Figure 83. Comparison of the accuracy (R^2) of the predicted temperature (top) and humidity (bottom) using the automatic model updating system.

4.7. CHAPTER CONCLUSION

In this chapter, the use of data-based modeling for the prediction of environmental changes in greenhouses was investigated. Based on the results, the NARX and RNN-LSTM models, which employ time series-based algorithms, performed better than the CNN-based model. This indicates that, in a greenhouse, more dynamic modeling is advantageous for predicting continuous and repetitive environmental changes. This finding is consistent with previous reports (Moon et al., 2018; Pawlowski et al., 2017).

For temperature, the prediction performance was very satisfactory, and the CO_2 concentration was also accurately predicted by all three models (ANN, NARX, RNN-LSTM). However, the overall prediction accuracy for humidity was very low for all models. The humidity prediction model reported by He and Ma (2010) produced an RMSEP of 1.67 using a combination of ANN and PCA models, which was better than that obtained in the present study using the RNN-LSTM model. Using the RNN-LSTM model, the humidity prediction performance was low, specifically from the afternoon of Day 5 to Day 6. During this period, the humidity was consistently higher than normal due to heavy rainfall. Further improvement in the humidity prediction models could be achieved by incorporating data from additional rain-detecting sensors into the training dataset. Furthermore, the greenhouse used a hydroponic system to provide nutrients and moisture to the roots, thus, adding irrigation history to the training data could also improve the humidity prediction results.

The shorter the prediction time, the greater the accuracy of the NARX model over general ANNs; however, the latter showed better performance at prediction times of over > 20 min. This suggests that time-series models are dependent on the current target values. In addition, the ANN model exhibited a relatively constant prediction performance irrespective of the time. On the other hand, although the RNN-LSTM model exhibited the best prediction performance, its accuracy declined with time, which suggests that the model could be effectively employed in the prediction of greenhouse environmental conditions and in greenhouse environmental control. For

instance, a predicted value could be used as an influence coefficient in a P-band controller. Moreover, it could be used to optimize the actuator signals used as input variables for the model. Another implication of the developed model is its possible role as a soft sensor. The model that was developed to predict the environmental conditions inside the greenhouse used external climate information and the history of the controller as input variables. In other words, the model can predict the environment inside the greenhouse without measuring it directly. This form of data sensing has been referred to as a type of soft sensor (Gonzaga et al., 2009; Sánchez et al., 2012).

The ANN, NARX, CNN-LSTM, and RNN-LSTM models were developed to determine the best approach to predicting changes in temperature, humidity, and CO_2, concentration, all of which directly affect the growth of greenhouse crops. The RNN-LSTM model exhibited the highest overall accuracy for temperature and CO_2 prediction with a SEP lower than 5% and an R^2 of 0.81−0.96, respectively. However, the accuracy of its humidity predictions was low; therefore, there was a need to improve this. Various training conditions were analyzed and the prediction performance for a time range of 5−30 min was determined. It was found that the accuracy of the time-based algorithm gradually decreased as the prediction time increased. However, the accuracy of the RNN-LSTM model was the highest at 30 min and, the accuracy converged after around 10 days of training data, which suggests 10 days is an appropriate threshold for field applications. The results of this study clearly show the potential for the use of deep-learning-based prediction models in precision greenhouse control. In particular, the CNN-LSTM model shows great promise for the prediction of environmental changes in greenhouses.

Finally, to develop a low-cost automatic learning model, an automatic environment model update system based on an embedded board was employed in an actual greenhouse. After 12 days of training, the automatic models exhibited a performance similar to that of models developed offline. The CNN-LSTM model in particular demonstrated excellent performance in the prediction of temperature changes.

5. DEVELOPMENT OF CLIMATE CONTROL SYSTEM BASED ON AI

Greenhouses are highly nonlinear and strongly coupled systems that are highly influenced by weather and the behavior of the actuators used for climate control (Zeng et al., 2012). In recent years, many studies have proposed control methods for greenhouse environments (Coelho et al., 2005; Fitz-Rodríguez et al., 2010; Piñón et al., 2005). Modern greenhouses use multiple-paned windows for more active natural ventilation, but it is more difficult to automatically control these complex systems and determine their effect on ventilation. Models of the window-opening process must consider both the external temperature and the effects of the wind direction and speed. In addition, it is difficult to continually develop new models for ever-changing configurations of greenhouses and their windows. Therefore, it is necessary to develop an improved ventilation control modeling method that can overcome these issues.

Many studies have designed ventilation control systems using models based on physical phenomena monitored within the greenhouse (Han et al., 2019; Hong and Lee, 2014; Montoya et al., 2016; Federico Villarreal-Guerrero et al., 2012). A similar approach attempts to guide ventilation-based changes in the greenhouse environment based on the weather or other relevant environmental factors (Benni et al., 2016b; Francik and Kurpaska, 2020; Rouphael et al., 2016). These methods use control logic based on the modeling of material properties according to the conservation of physical energy or the experience-based empirical modeling of greenhouses.

The most realistic control algorithms are based on PID logic, but it is difficult to apply this effectively in complex systems such as greenhouses because the relevant coefficients must be tuned (Chen and Huang, 2004). Furthermore, it takes a long time for the actuator to affect the internal environmental variables such as the internal temperature and humidity, and the influence of other environmental factors is considerable. Therefore, most studies have been conducted using simulations (Rodríguez et al., 2015; Zeng et al., 2012), and highly sophisticated model development is needed to tune the relevant coefficients based on a variety of physical phenomena.

Models based on neural networks are suitable for both linear and nonlinear modeling and have been applied to greenhouse environment modeling and control logic (He and Ma, 2010; Jung et al., 2019; Taki et al., 2018). Many studies have reported reliable results in environmental prediction modeling using ANNs (He and Ma, 2010; Khashei and Bijari, 2010; Li et al., 2007; Sukhatme et al., 2007; Teófilo et al., 2009). However, there have been few reports of this method being used to control greenhouse environments, with most having focused on simulations. For example, Fourati and Chtourou (2007) adopted an Elman neural network to emulate the dynamics of simulated greenhouse performance using a neural-network-based controller. Similarly, Fitz-Rodríguez et al. (2010) designed a dynamic greenhouse environment simulator for use in testing greenhouse control principles.

The modeling of greenhouse environmental changes has led to studies investigating the use of predictive control (Blasco et al., 2007; Coelho et al., 2005; Ramos Ruiz et al., 2019; Zeng et al., 2012). For example, Blasco et al. (2007) assessed model-based predictive control logic and an optimization technique based on a genetic algorithm, with results showing that there is significant flexibility in selecting the control objectives. In addition, Coelho et al. (2005) applied a particle swarm optimization algorithm to control logic in a greenhouse air temperature controller and computed outputs to optimize the future environment of the greenhouse.

Techniques that utilize input-output data processing based on ANNs are referred to as black-box modeling (Rodríguez et al., 2015; Taki et al., 2016). However, these models need to be understood and interpreted to determine the parameters for the model-based controller, and black-box models need to be analyzed empirically. In order to determine the parameters of these model-based controllers, it is necessary to understand and interpret the model (Kim et al., 2019). For optimal nonlinear tracking, output feedback neural-network (OFNN)-based techniques have been developed to generate an approximate solution (Ding et al., 2019). Due to the inherent approximation capability of neural network systems, adaptive neural controllers have been proposed for nonlinear systems (B. Chen et al., 2015; Li et al., 2009; Theodorakopoulos and Rovithakis, 2014). The possibility of applying neural-network-based control algorithms to greenhouse environments, where mathematical modeling is difficult, has been reported (Blasco et al., 2007), but the practical

applications remain underexplored. Therefore, the specific aims of this chapter are to

1) Use the OFNN structure to propose a method for optimizing the control signal with the deep-learning climate prediction model developed in Chapter 4
2) By comparing various optimization methods, develop a structure suitable for greenhouse climate control and confirm the feasibility of the proposed logic by applying it to actual single-span greenhouse ventilation control.

5.1. OUTPUT FEEDBACK NEURAL NETWORK (OFNN) MODEL FOR OPTIMAL CONTROL

Many previous studies have applied AI prediction models and deep-learning techniques for changes in the greenhouse environment. However, in terms of utilizing these models, specific research directions are currently lacking (Theodoridis, 2015). In other words, the environment needs to be predicted and the subsequent management or treatment response should be established. This chapter describes an approach to this and proposes related techniques. Deep-learning and neural-network-based system models are highly nonlinear. In order to reinterpret this or obtain an optimized value, a nonlinear optimization technique based on gradient descent should be utilized based on the trial-and-error method (Qian, 1999).

5.1.1. PERFORMANCE COMPARISON BY OPTIMIZATION MODULE

The OFNN structure was demonstrated in the previous chapter, illustrating that gradient-descent-based techniques can be employed as a module to explore a black-box model and find the optimal solution (Figure 84 and Figure 85). Currently, many stochastic gradient descent (SGD)-based parallel learning algorithms apply various approximation methods to improve computational speed. SGD is the most widely used backpropagation learning method for DNNs. An optimization algorithm based on SGD is inherently sequential, making it difficult to parallelize. A widespread strategy is to remove or ignore this sequential processing method to improve the learning speed by processing the gradient value calculated by SGD using an approximate method.

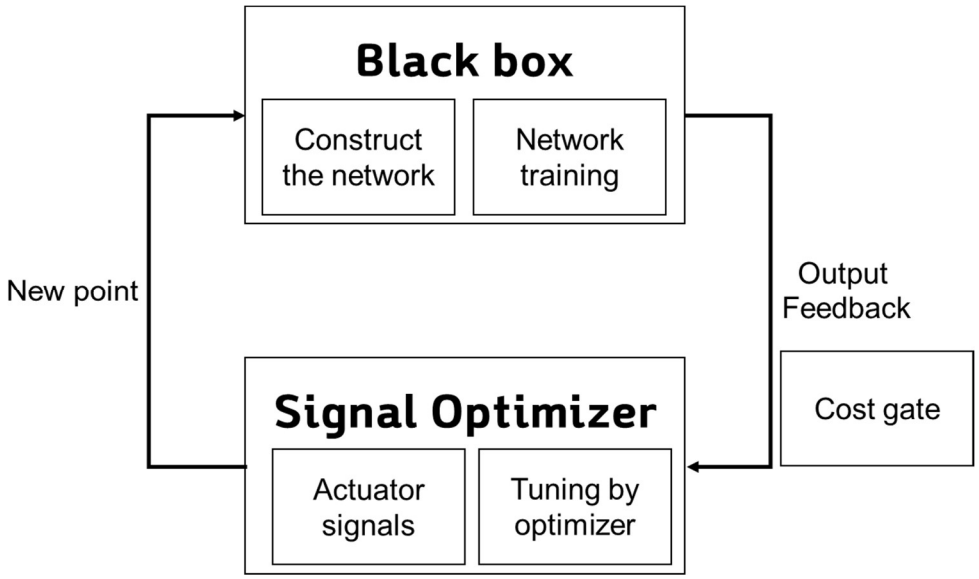

Figure 84. Framework for determining the optimal signal in a deep-learning model.

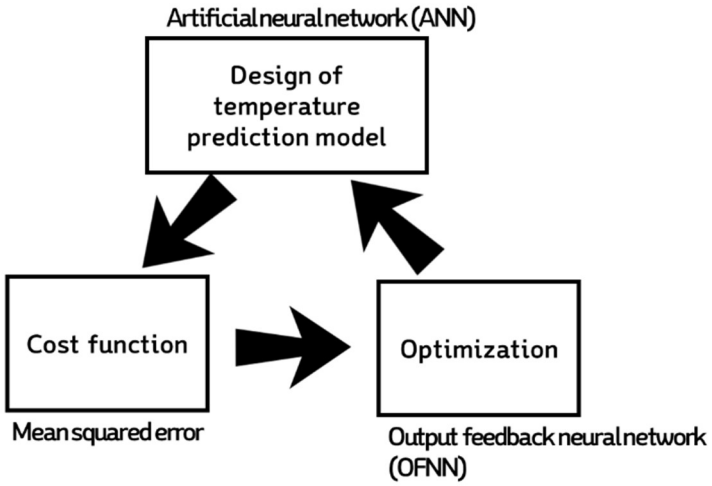

Figure 85. Schematic of the neural-network-based temperature prediction model and optimizer.

Stochastic gradient descent

Various methods have been employed to solve this sequential method and to parallelize the learning algorithm. Currently, a widely used learning method for DNNs is SGD-based backpropagation. The OFNN search method can be expressed as an optimization problem in finding parameters that minimize *f*, the multivariate objective function (Equation [39]).

$$f(\sigma[W,x],y)min_W \ , W = [w_1, w_2, \cdots w_n] \quad \cdots [39]$$

Here, *w* is a model parameter, *x* is a part or all of training data *X* = [x1, x1, ⋯ xi], and σ is an activation function. To solve this optimization problem, the following gradient descent is repeatedly conducted to optimize the parameters of the OFNN. The most basic structure of gradient descent is as follows:

$$w_{t+1} = \ w_t - \eta \ \odot \ \nabla f(w_t; x_t; y_t) \quad \cdots [40]$$

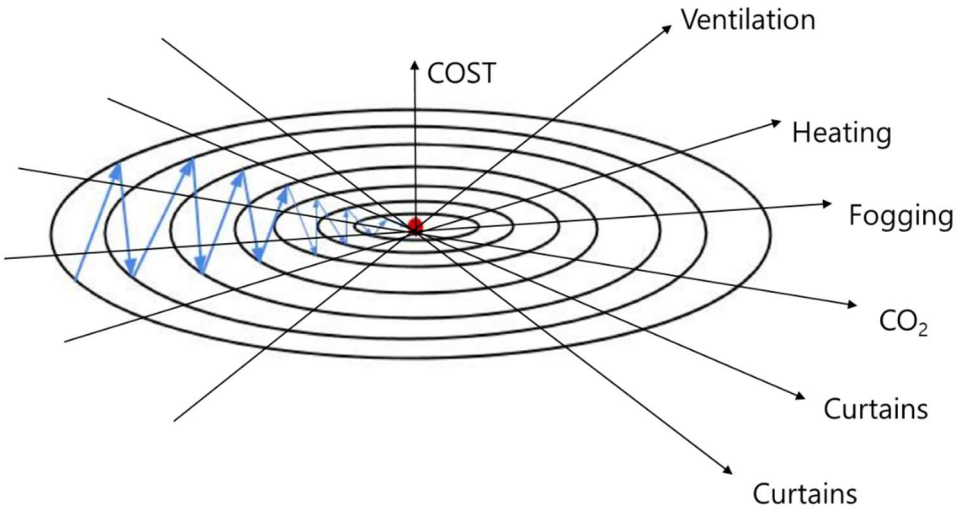

Figure 86. Oscillating Gradient Descent

Figure 86 presents the optimized route taken by regular gradient descent from the starting point to the minimum point, which takes a zigzag form. The algorithm could converge earlier if the values for the oscillating component is reduced while maintaining the horizontal component. The algorithms SGD with momentum, RMSprop, and Adadelta are employed to address this issue, and their performance is compared for real greenhouse control history data.

SGD with momentum

SGD has difficulty navigating ravines, which are areas where the surface curves much more steeply in one dimension than in another (Qian, 1999); these are particularly common around local optima. In this scenario, SGD oscillates across the slope of the ravine while making slow progress towards the local optimum. Momentum can be used to accelerate SGD in the desired direction and dampens oscillations (Figure 86). It does this by adding fraction γ of the update vector of the previous time step to the current update vector as follows:

$$\upsilon_t = \gamma\upsilon_{t-1} + \eta\nabla_\theta J(\theta) \quad \cdots [41]$$

$$\theta = \theta - \upsilon_t \quad \cdots [42]$$

$J(\theta)$: an objective function

θ : current parameter

$\gamma \upsilon_{t-1}$: momentum term

Adadelta

Zeiler (2012) presented a novel per-dimension learning rate method for gradient descent called Adadelta. This method dynamically adapts over time using only first-order information and has minimal computational overhead beyond vanilla SGD. The method requires no manual tuning of the learning rate and appears robust to noisy gradient information, different model architecture choices, various data modalities, and the selection of hyperparameters. Adadelta is an extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Instead of accumulating all past squared gradients, Adadelta restricts the window of accumulated past gradients to fixed size ω in Equation [43]:

$$\mathrm{E}[\mathrm{g}^2]_t = \gamma \mathrm{E}[\mathrm{g}^2]_{t-1} + (1-\gamma)\mathrm{g}_t^2 \quad \cdots [43]$$

Instead of inefficiently storing ω previous squared gradients, the sum of the gradients is recursively defined as a decaying average of all past squared gradients. The running average $\mathrm{E}[\mathrm{g}^2]_t$ at time step *t* then depends (as fraction γ depends similarly on the momentum term) only on the previous average and the current gradient. The parameter update vector of Adagrad that was derived previously thus takes the form [43] to [44]:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + e}} \odot g_t \quad \cdots [44]$$

Because the denominator is just the RMSE criterion of the gradient, this can be replaced with a short-hand criterion:

$$\Delta\theta_t = -\frac{\eta}{RMS[g]_t} g_t \quad \cdots [45]$$

The update should have the same hypothetical units as the parameter:

$$E[\triangle \theta^2]_t = \gamma E[\triangle \theta^2]_{t-1} + (1-\gamma) \triangle \theta_t^2 \quad \cdots [46]$$

The RMSE of the parameter updates is thus

$$\mathrm{RMS}[\Delta\theta]_t = \sqrt{E[\triangle \theta^2]_t + \epsilon} \quad \cdots [47]$$

$\mathrm{RMS}[\Delta\theta]_t$ was approximated by the RMSE of the parameter updates until the previous time step. Replacing the learning rate η in the previous update rule with $\mathrm{RMS}[\Delta\theta]_{t-1}$ finally yields the Adadelta update rule:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t \quad \cdots [48]$$

$$\Delta\theta_{t+1} = \theta_t + \Delta\theta_t \quad \cdots [49]$$

RMSprop

RMSprop is an adaptive learning rate method that has gained popularity in recent years but which has also received some criticism (Wilson et al., 2017). It is famous for not yet having been published but being very well-known regardless. Indeed, most deep-learning frameworks include an implementation of this out of the box. RMSprop and Adadelta were both developed independently around the same time stemming from the need to resolve Adagrad's radically diminishing learning rates. RMSprop is identical to the first update vector of Adadelta described above:

$$\mathrm{E}[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad \cdots [50]$$

$$\Delta\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t \quad \cdots [51]$$

RMSprop combines the idea of only using the sign of the gradient with the idea of

adapting the step size individually for each weight rather than looking at the magnitude of the gradient.

5.1.2. EXPLORING GRADIENT DESCENT-BASED OPTIMIZERS FOR LINEAR REGRESSION EXAMPLE

This case study explores the simplest possible supervised learning algorithm: linear regression. This case study refers to the study of the previous study (Ruder, 2016). Doing so allows the shortcomings of different optimizers to be clearly understood, and these issues can be overcome using newer methods. Optimizers are used to optimize the model parameters of learning algorithms. They basically differ in their updating step given the current parameter value and the gradient value for all parameters. Updating the parameters to minimize the loss function is very similar to the example of a person trying to reach the bottom of a valley. At any point, they know where they are, and the direction of maximum change (gradient) in their neighborhood (Ruder, 2016; Theodoridis, 2015). The updated point keeps moving in the direction of the maximum downward slope until the algorithm finds that the slope moves upwards in all directions.

$$\mathrm{y} = 5\mathrm{x} + 55 + \mathrm{w},\ \textit{where}\ w \in N(0, \sigma^2) \qquad \cdots [52]$$

It is a line with some Gaussian noise added. The true value of the parameters of the line, i.e. *a* and *b*, are known. In a supervised learning problem setting, the task would be to determine *a* and *b*. Starting with an initial pair (a, b), the direction of the optimizer is followed until the value of (a, b) that minimizes the mean squared error over all data points (i.e., the global minima) or a batch of data points (i.e., the local minima) is found:

$$J = \frac{1}{2} \sum_{i=1}^{n} (y_i - (ax_i + b))^2 \qquad \cdots [53]$$

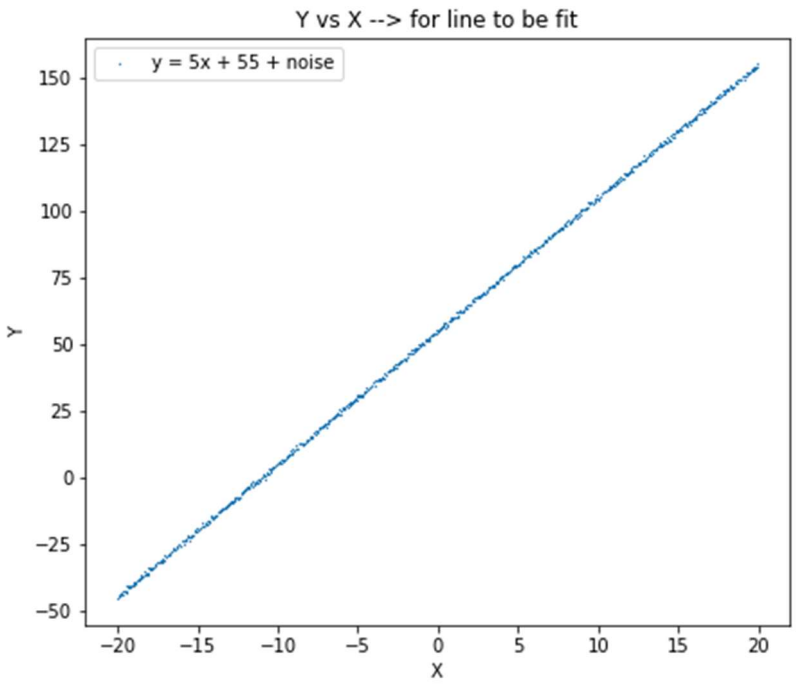

Figure 87. Simple linear regression equation with noise used for case study.

In Figure 87, the blue line at any point is the loss for a particular set of parameters (a, b). Notice how the rise or fall in the loss is much steeper when changing *a* than when changing *b*. This is even more evident in Figure 88 to Figure 91, in which the optimization process, fluctuation, and speed can be confirmed for each feature of the GD algorithm.

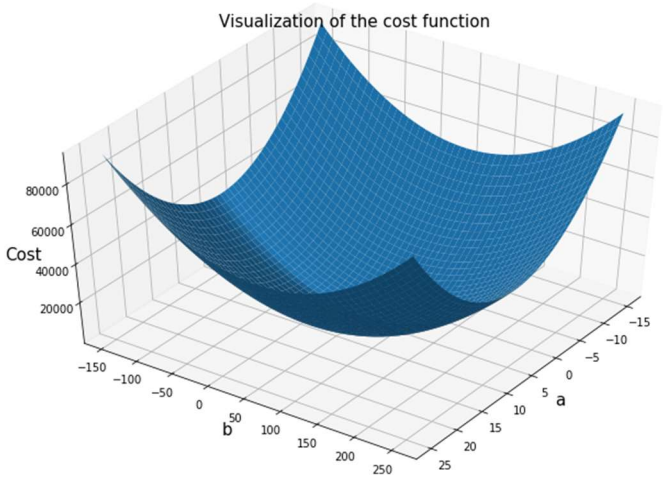

Figure 88. 3D graph with visualized cost function.

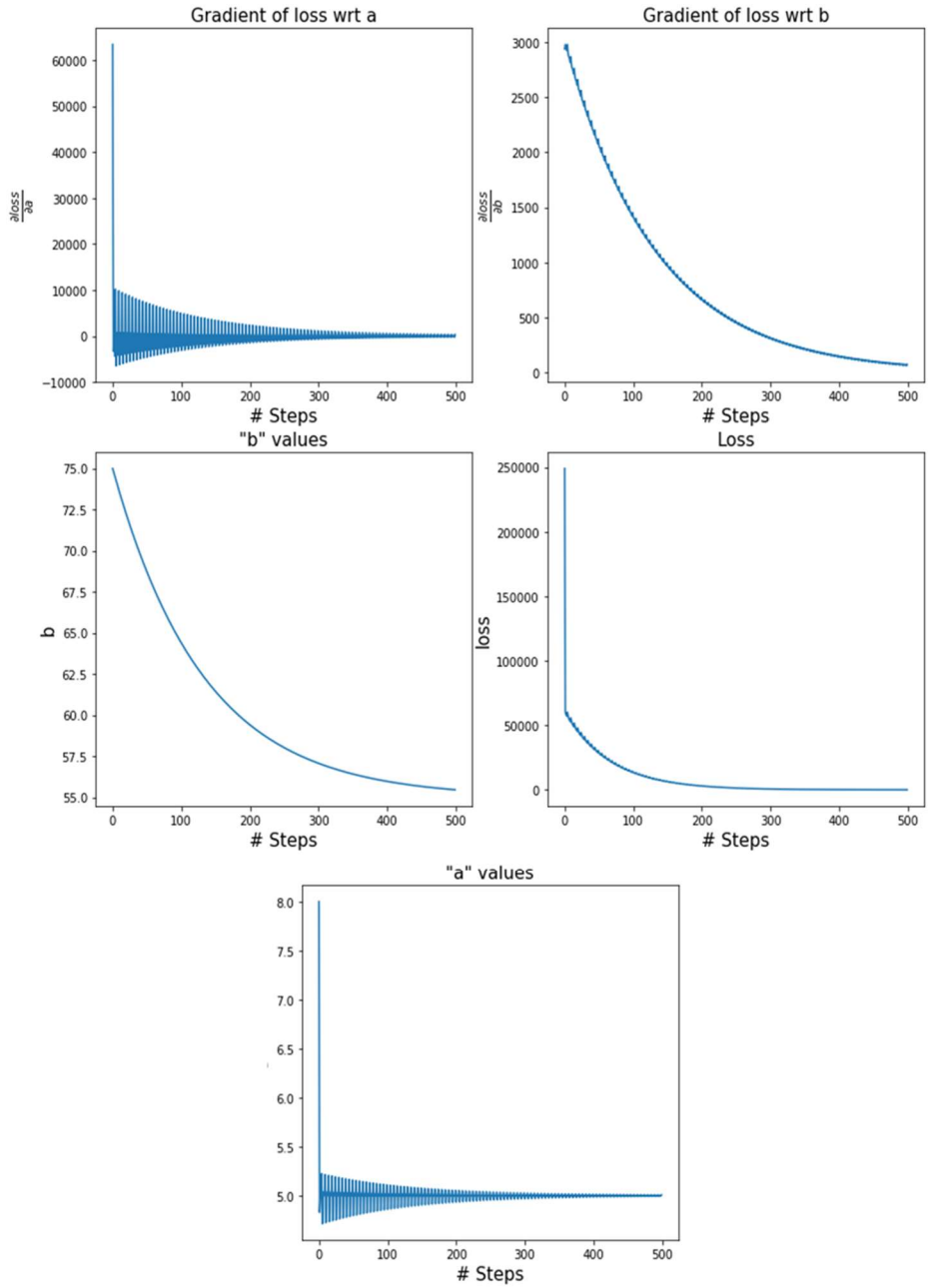

Figure 89. Loss change for each step and optimization coefficient change in SGD algorithm (a,b).

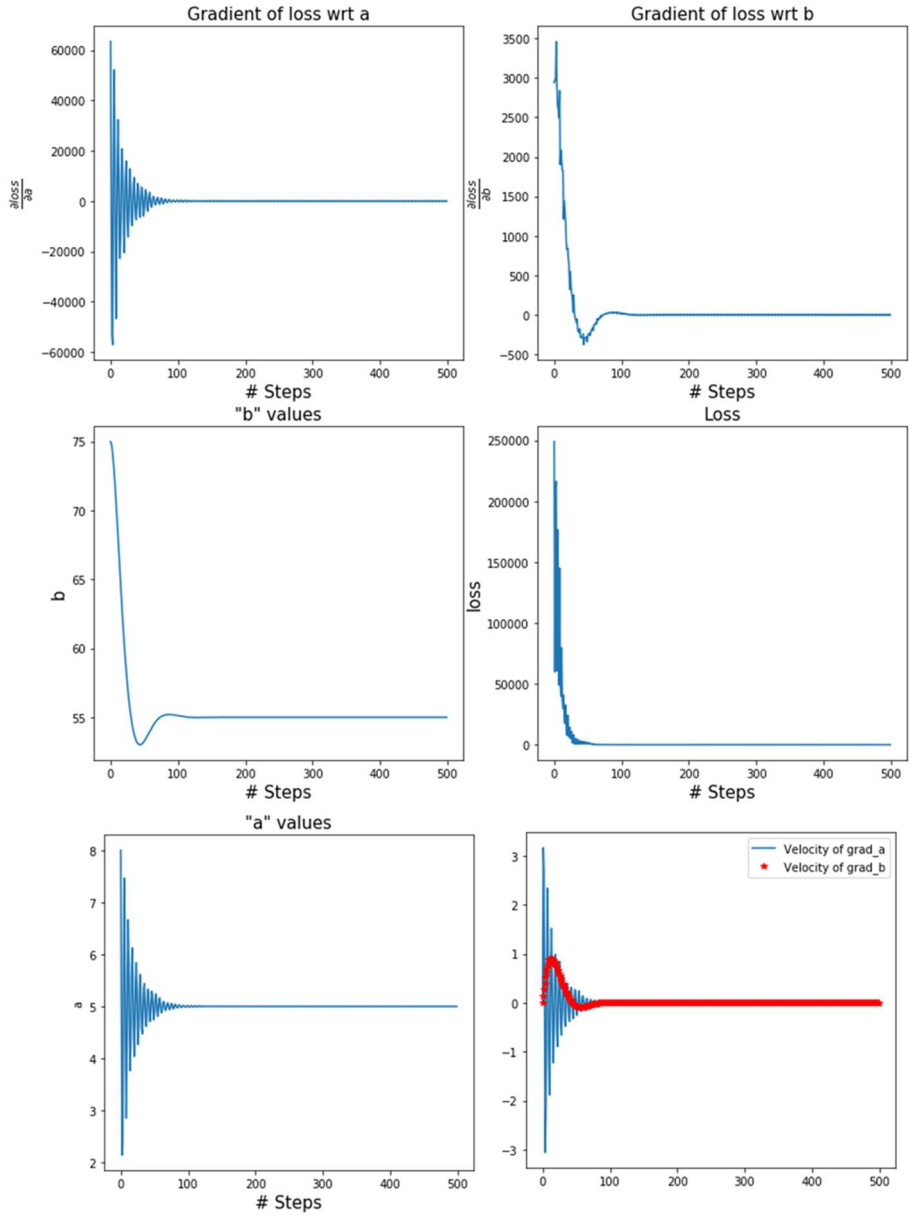

Figure 90. Loss change and optimization coefficient for each step in the Momentom algorithm (a,b) and change in the velocity coefficient value.

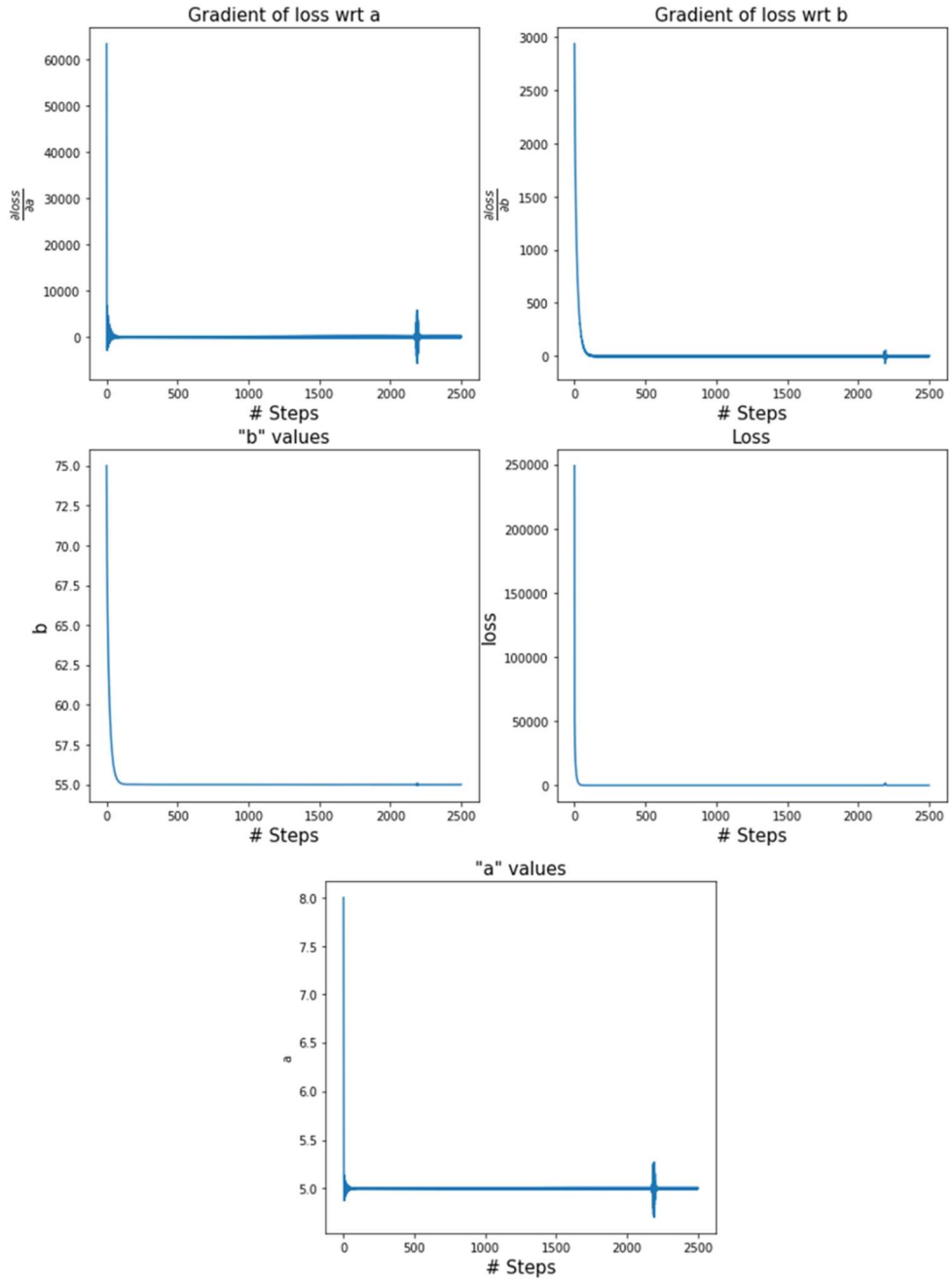

Figure 91. Loss change and optimization coefficient for each step in the adadelta algorithm (a,b).

5.1.3. GREENHOUSE CONTROL SIGNAL DETERMINATION FOR PERFORMANCE COMPARISON OF OPTIMIZATION ALGORITHM

The three optimization algorithms (SDG, SDG with momentum, and Adadelta) have different characteristics, so OFNN logic consisting of actual data and a prediction model was employed to identify which algorithm was best-suited to determining greenhouse control signals. In this study, three optimization algorithms were applied. By applying the SGD, SGD with momentum, and Adadelta optimization algorithms, the number of iterations required to find the optimal point was compared.

Simulation of ventilation windows

The algorithms were first applied to optimize the control model for opening the ventilation windows to control the temperature. On September 25, 2019, a sharp rise in temperature was observed from 9 am to 10 am. At around 9:20, the set target temperature was exceeded. At this time, the window open command required an opening of 58% for the left windows and 41% for the right (Figure 92). This was a result of considering the external climate and the internal temperature of the greenhouse in the P-band control. In this study, the OFNN algorithms were used to determine the optimal ventilation at this point.

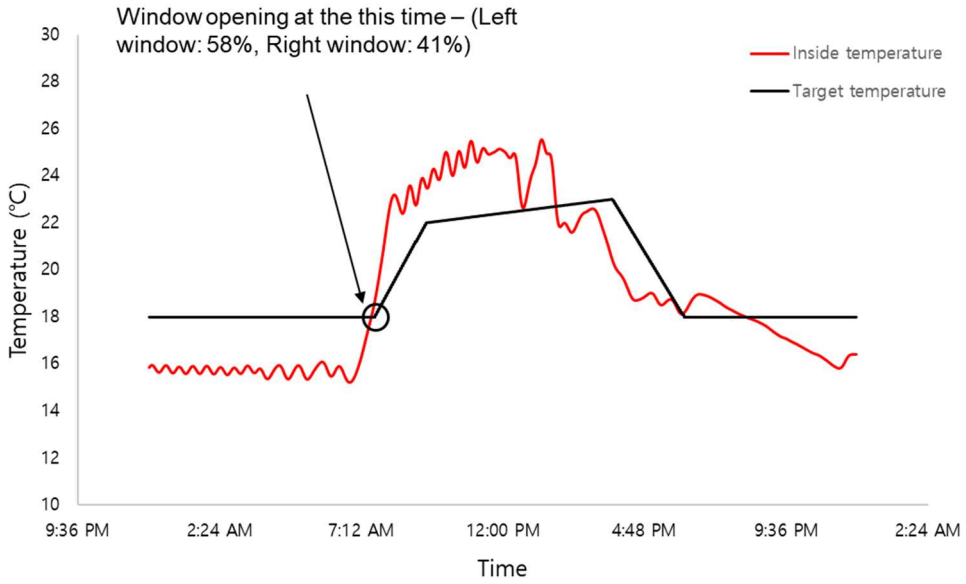

Figure 92. Temperature profile for the optimization algorithm simulation.

The OFNN is structured as shown in Figure 85. Using the predictive model and the output values obtained from the predictive model, the cost was calculated, and the SGD-based algorithm was applied to optimize the cost. As a result, the change in cost for the opening of the left and right windows is presented as a contour map in Figure 93.

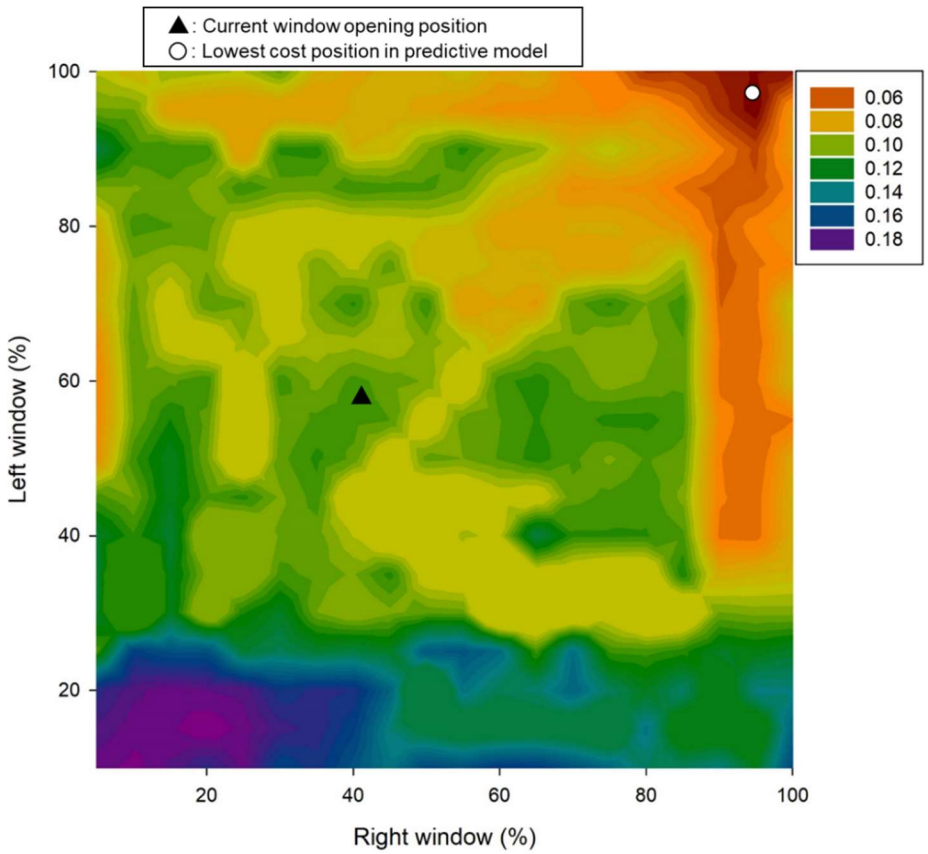

Figure 93. Contour map of the change in cost according to the opening of the left and right windows in the simulation.

Table 30. Predicted values for the temperature model according to the opening position of the windows (corresponding cost in parentheses)

Left \ Right	10%	20%	50%	80%
10 %	19.21 (0.252)	19.16 (0.217)	19.02 (0.135	18.98 (0.115)
20%	19.12 (0.192)	19.11 (0.186)	18.98 (0.115)	18.97 (0.11)
50%	19.07 (0.162)	18.99 (0.120)	18.92 (0.082)	18.85 (0.075)
80%	18.99 (0.120)	18.92 (0.088	18.98 (0.115)	18.72 (0.0242)

To generate the cost contour map, the left and right windows were input in steps of 2.0%, generating about 2,500 output results. The x-axis represents the opening of the right window and the y-axis represents the opening of the left window. The current position of the left and right windows is indicated by the (▲) symbol in the center of the figure. The point at which the cost becomes the lowest is (○), converging at an open signal of 99% for the left window and 98% for the right window.

Figure 94 presents the results for the optimized search path using SGD. A total of 88 iterations were conducted, with each triangle representing two iterations. Due to the nature of SGD, it was found that the slope search tended to be done in a certain step, thus the search fell into a local minimum. Past research has indicated that this can be solved using a momentum-based algorithm.

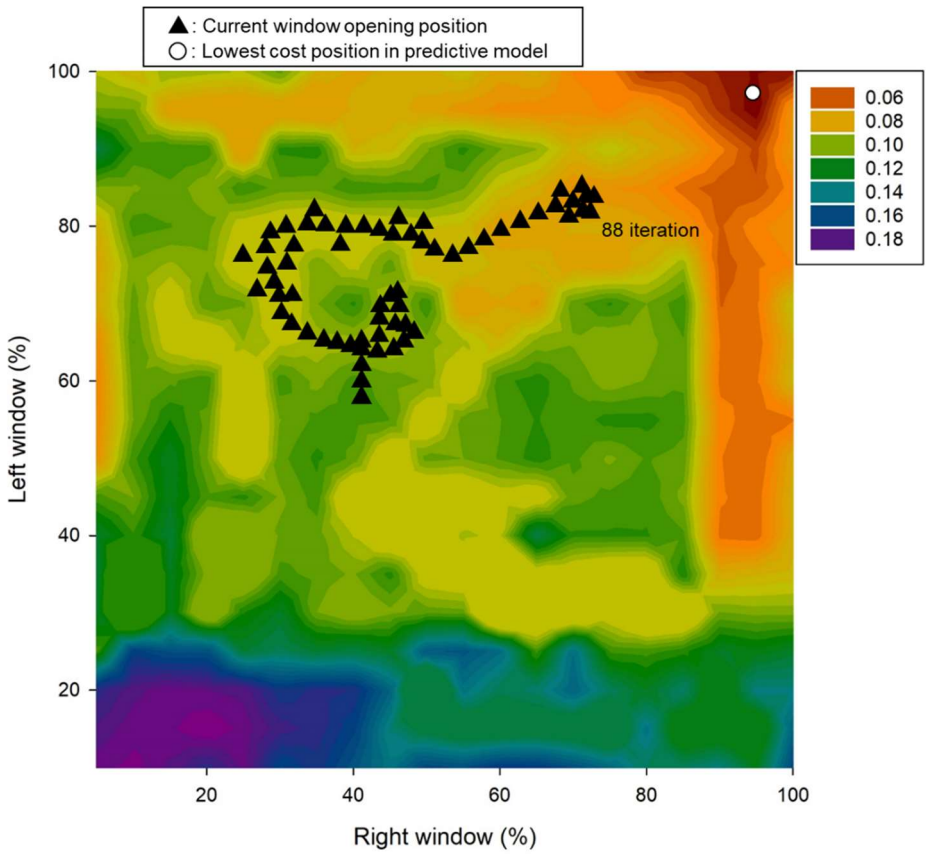

Figure 94. Contour map of the change in cost according to the ventilation rate and the trajectory of the change in the optimization point with SGD.

Figure 95. shows the results of the search for the optimization point using the momentum-based algorithm. Unlike SGD, the search trajectory moved past the local minimum through an accelerated route. The minimization point was found after 33 iterations. In this case, it was decided that the opening of the right window was more favorable for minimization. This is because the wind at the time was blowing in an upward direction, which gave more weighted calculation to the direction in which the wind was blowing to the left-right opening degree in the existing P-band-based control.

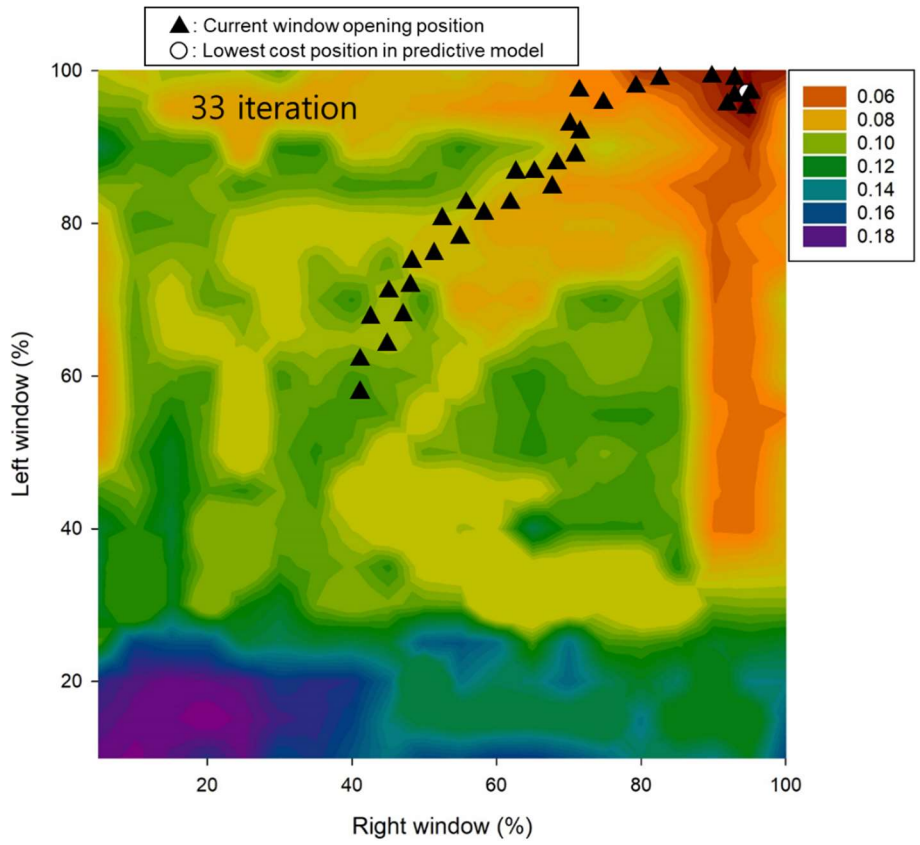

Figure 95. Contour map of the change in cost according to the ventilation rate and the trajectory of the change in the optimization point with SGD

Figure 96 presents the optimized results using RMSprop. The optimal point was reached over 50 times, and the path was similar to that observed for SGD with momentum, but the number of iterations was lower (28). Figure 97 presents the optimized route using Adadelta, for which an optimal conclusion was obtained after 24 iterations. Adadelta is a logic that combines the RMSprop's exponential moving average function, Adadelta's optimal path tracking theory, and the advantages of SGD with momentum. It thus provided the best performance here and has been actively used to train many deep neural networks in past research.

In this study, the signal determination process for opening the windows for ventilation control was explored. It is expected that the opening of the windows will be more optimal if the prediction model is able to accurately detect environmental changes within the designed OFNN structure. However, because these are only simulations results, further research is required to determine the control signals in

real-time through field applications.

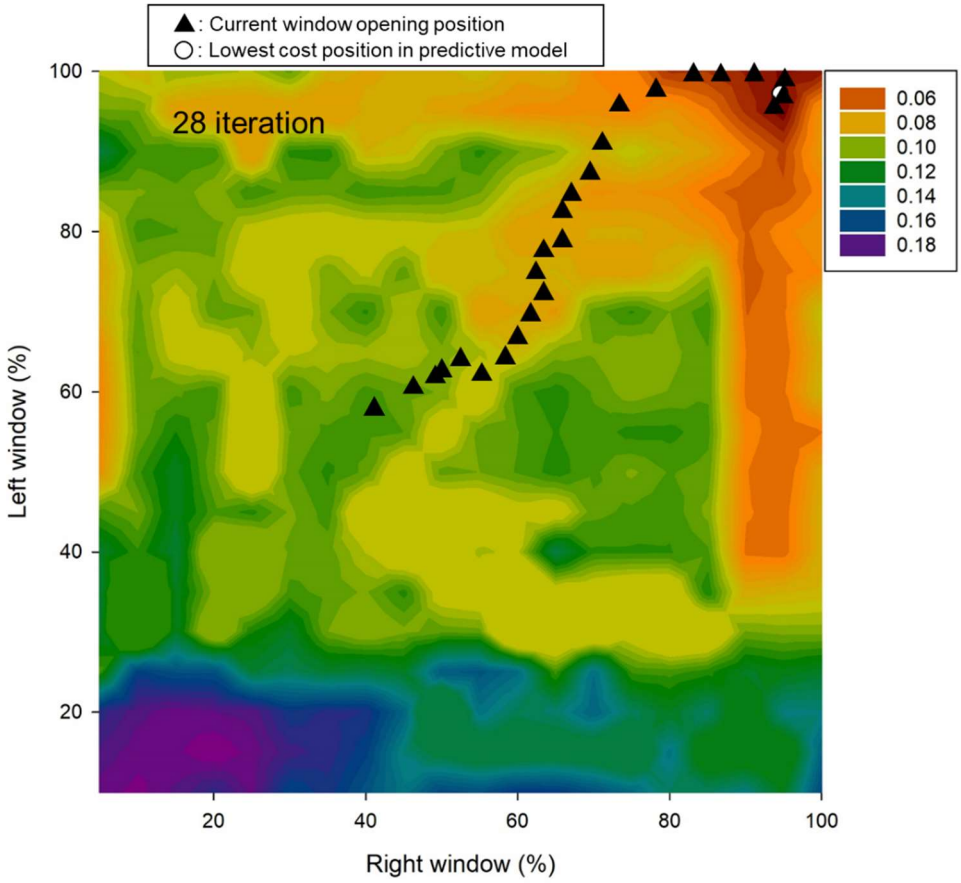

Figure 96. Contour map of cost change according to ventilation rate and trajectory of optimization point change with RMSprops.

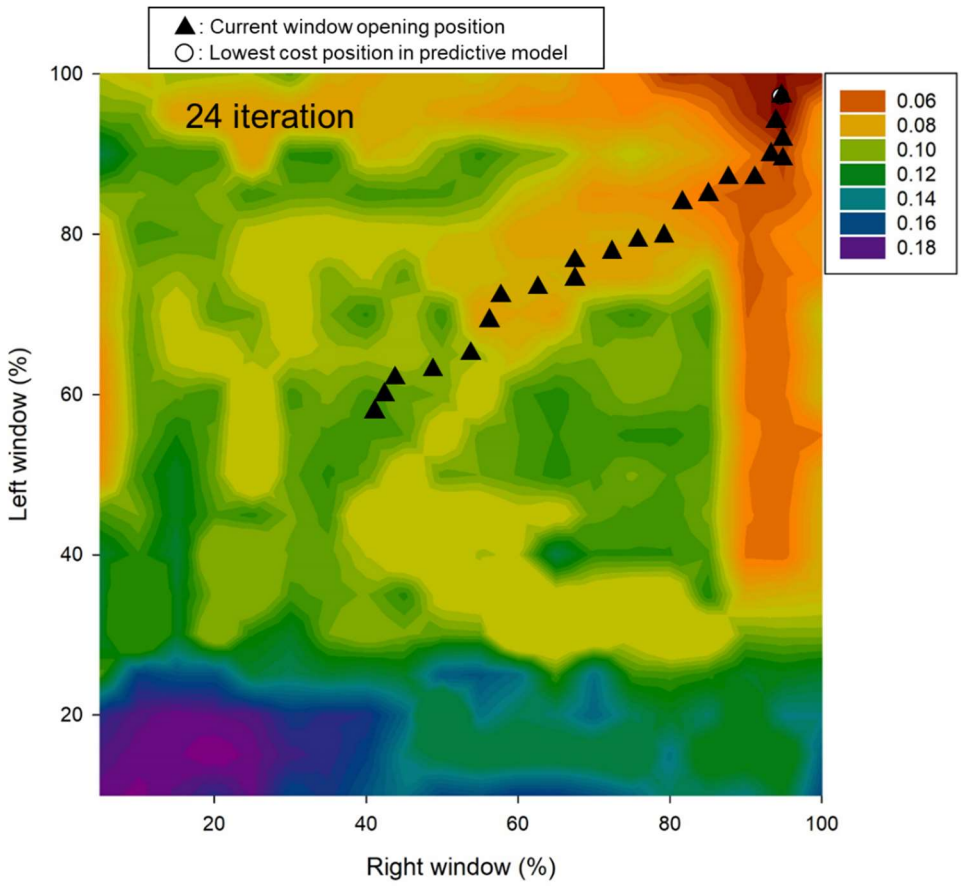

Figure 97. Contour map of cost change according to ventilation rate and trajectory of optimization point change with Adadelta.

5.2. OUTPUT FEEDBACK NEURAL NETWORK (OFNN) APPLICATION FOR OPTIMAL VENTILATION CONTROL

As the modern systems to be controlled continue to become more complex, the control theory of uncertain nonlinear systems has also been developed (Tzirkel-Hancock and Fallside, 1992). It is often difficult to determine a mathematical model for a complex system. Erroneous modeling or a simplified model implies non-structured uncertainty that is not found in the real systems. In order to respond to such a case, the intelligence control theory for introducing neural networks into controllers has been actively developed (Calise et al., 2001; Zhang and Wang, 2001).

The primary goal of this study is to develop a precise model of predicting the environmental change of greenhouse through neural network systems. It is also necessary to determine a control logic that is suitable to respond to the predictions of environmental change. The control logic of a strongly nonlinear system like the environment of greenhouse presents only a state variable feedback controller under the assumption that every state variable can be observed by an adaptive control system. As, in a real greenhouse, only a few state variables can be measured, optimization at the output part is practically more advantageous. This type of control has not been dealt with in any of the existing studies.

In Addition, this chapter focused on developing a new approach to applying control logic based on an OFNN algorithms for improved control of natural greenhouse ventilation. This is a promising idea based on a model that predicts temperature changes in response to window opening activity in greenhouses. In order to validate proposed method, the method was applied to a real greenhouse and verified by comparing the results. The next section describes experimental greenhouse, the design of the proposed method.

The optimization of control is a kind of neural network that determined and outputs the on-off control signals of multiple roof and side windows for optimal ventilation by using a cost function about the difference between a predicted environmental value obtained by an environmental prediction model and a target temperature. As shown in Figure 98, the optimization of control includes a feedback neural network that has multiple optimization nodes producing control signals of multiple

controllers. Based on the cost function values thus calculated (Figure 98), the optimization of control utilizes the gradient descentmethod to update the weights of multiple optimization nodes and thus determines control signal values of roof and side windows. Specifically, the weights of multiple nodes continue to be updated by reducing cost through the partial differentiation of cost with respect to the weight of the optimization node.

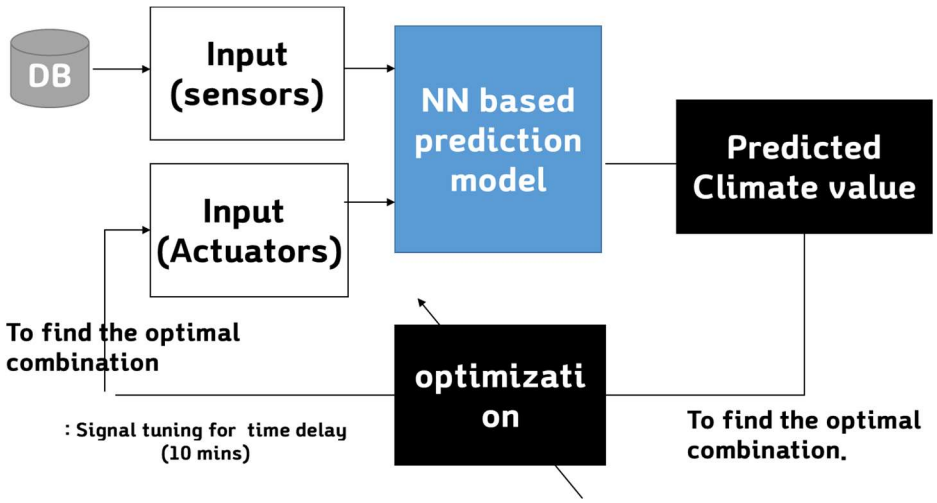

Figure 98. Conceptual diagram of output feedback neural network-based greenhouse environmental control

5.2.1. NEURAL-NETWORK-BASED TEMPERATURE PREDICTION MODEL

Our proposed temperature control method was theoretically founded on model-based predictive control logic (Figure 98). First, a prediction model was developed based on data accumulated in the database, mainly sensor data and actuator history. The prediction model was a black box model based on an artificial neural network; the output value was the predicted inside temperature after 30 min. The control decision determines the control signal value corresponding to the input of the prediction model feedback from the output of the prediction model to the optimization node. The ANN-based prediction model consisted of four layers of neurons or nodes: the input layer, two hidden layers, and the output layer (Figure). Signals entering the input layer are transmitted to the hidden layers and output layer

through functions. The model used 15 input variables divided between the data obtained from the sensors and the current controller history. The first hidden layer used 45 nodes and the hyperbolic tangent (Tanh) activation while the second hidden layer used 30 nodes and the Rectified Linear Unit (ReLU) activation function (Table 31). In the previous study, the results were verified through various combinations of active functions in the hidden nodes and parameter values, but no significant correlation was found. For training, Levenberg–Marquardt algorithm was used, a gradient descent method for avoiding local minima and overfitting (Jung et al., 2019) of the 73,440 samples used for model development, training used 70%, validation 15%, and testing 15%.

Table 31. Functions used in the artificial neural network hidden nodes.

Name	Equation	Derivative
Tanh	$f(\mathrm{x}) = \frac{2}{1+e^{-2x}} - 1$	$f'(\mathrm{x}) = \frac{1}{x^2+1}$
ReLU	$f(\mathrm{x}) = \begin{cases} 0 \ for \ x < 0 \\ x \ for \ x \ \geq 0 \end{cases}$	$f'(\mathrm{x}) = \begin{cases} 0 \ for \ x < 0 \\ 1 \ for \ x \ \geq 0 \end{cases}$

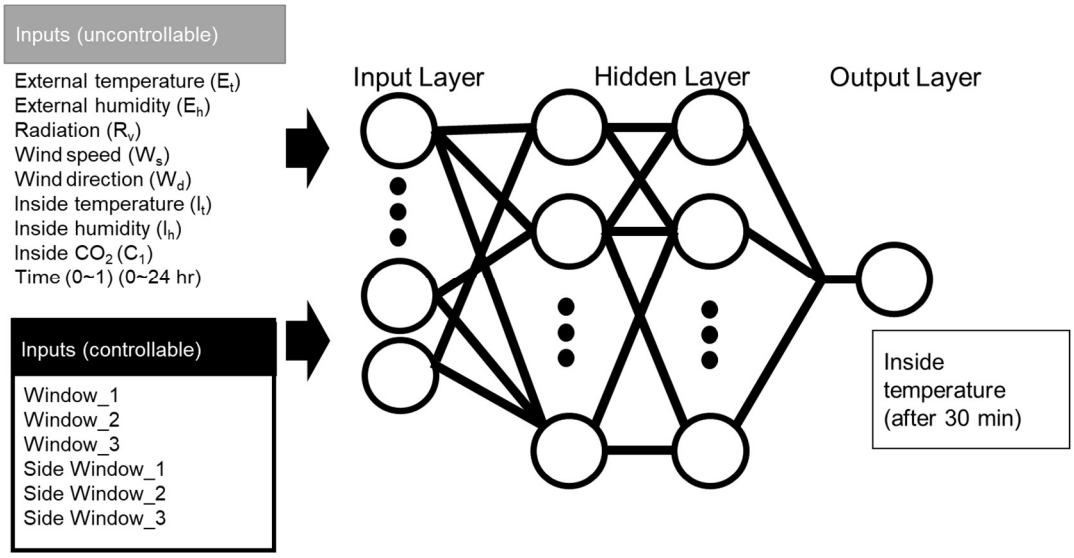

Figure 99. Structure of the neural network model for prediction of inside temperature.

Our ventilation control model was based on the neural network output feedback method, which does not affect the training process but operates through the prediction model already developed. The logic determining window opening for ventilation uses the momentum-based gradient descent method by setting a node with a separate sub-routine that calculates the difference between temperature change after 30 min and the target temperature. This operates through a cost function based on the mean square error (MSE):

$$C_m(k) = \frac{1}{2} e_m(k)^2 \qquad \cdots [54]$$

where $e_m(k)$ is the error between the target temperature and predicted model output.

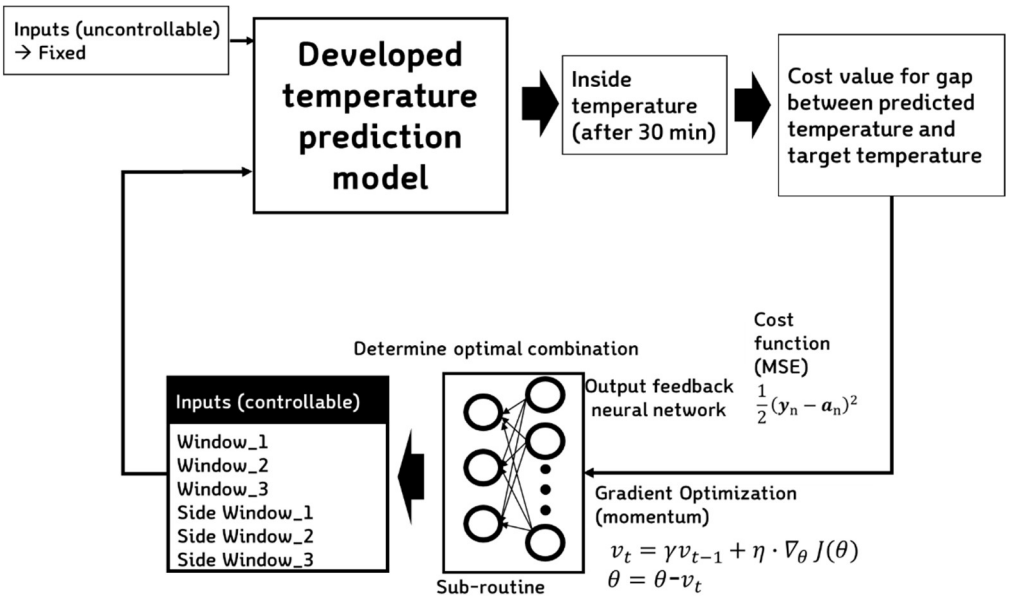

Figure 100. Structure of the output feedback neural network (OFNN) and operational direction.

The proposed logic operates to reduce em (k) through a separate artificial neural network layer installed to adjust the weight of the neural network nodes (Figure 4). In order to adjust the weight parameter, we included a momentum term, a well-known function used to increase the rate of convergence dramatically (Qian, 1999; Schmidhuber, 2015):

$$\Delta w_t = -\varepsilon \nabla_w E(W) + p\Delta w_{t-1} \quad \cdots [55]$$

where p is the momentum parameter (0.01 was used). The modification of the weight vector at the current time step depends on both the current gradient and the weight change of the previous step (Schmidhuber, 2015) .

The sub-routine stops repeating if three conditions are met (Figure 103): (1) when the cost function value calculated by the predicted temperature decreases below a certain value (Cost: 0.01), (2) when the progress value (difference between the value before and after the update) decreases below a certain value (r: 0.001), and (3) when the number of iterations reaches the set number of times (i: 100). At this point, the iterations stop and the process outputs the determined value to the signal

corresponding to the opening of six windows. The OFNN operates in conjunction with the predictive model, and the input variables are the MSE calculation results between the target temperature and the current temperature. The hidden node is composed of two layers, each having 15 and 10 nodes, and Tanh is used as the activation function. The output is six nodes, each of which directly determines the window open control signal.

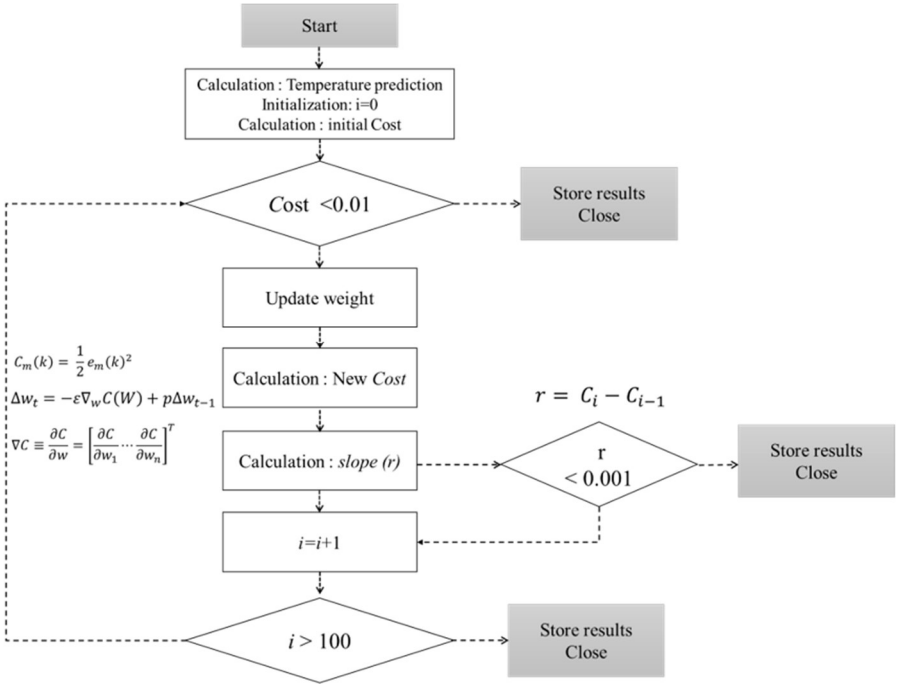

Figure 101. Flowchart for optimal control values using the gradient descent method.

5.3. APPLICATION OF OFNN TO VENTILATION CONTROL OF SINGLE SPAN GREENHOUSE

5.3.1. DESCRIPTION OF EXPERIMENTAL GREENHOUSE

The greenhouse used in this study is a strawberry growing greenhouse and exterior is made of polyvinyl. The upper part of the greenhouse is arched and consists of about 3 layers of wall, which is consist of a width of 7 m, a height of about 3 m, a length of 70 m and a total growing area of about 500m^2. This strawberry farm consists of three greenhouses with the same area as described above, each of which can be controlled independently. The interior and exterior photographs of the greenhouse are shown in Figure 102 (a) and (b). The greenhouse is equipped with an automated system, with sensors for monitoring the internal temperature, humidity and carbon dioxide, and automatically controls windows and internal fans for temperature control. In the present study, the ventilation control system installed in advance is designed to calculate the ventilation load using a linear algorithm and operates with a P (Proportional)-band based algorithm (Kim et al., 2017). The accumulated data is obtained by operating P-band control logic during two months, which has all of the environmental information and control history from March 2018 to May 2018.

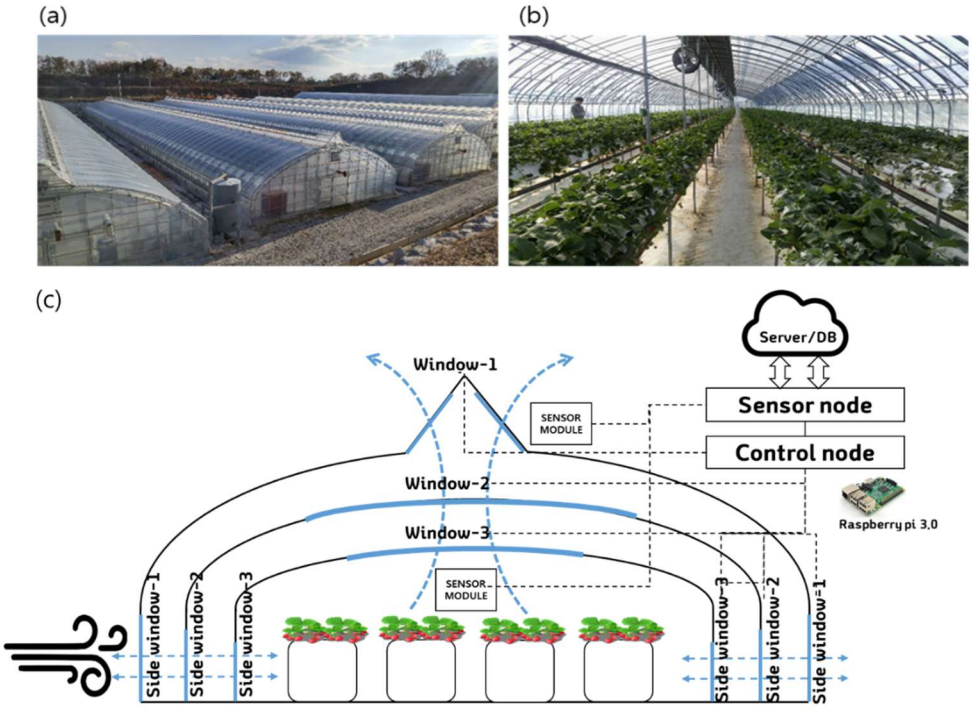

Figure 102. Strawberry greenhouse used in this study: (a) exterior and (b) inside with multi-window shell structure, and (c) schematic of experimental greenhouse monitoring and control system.

The configuration diagram of the greenhouse is shown in Figure 102 (c). Temperature and relative humidity and CO_2 concentration of inside climate were measured by using two sensor modules (SH-VT250, Soha tech, Korea), and the two sensor values were averaged. The sensors were installed at the center of greenhouse, and the specifications of the sensors are shown in table 1. The environment controller consists of a sensor node and a control node for processing sensor data (Figure 102 (c)). The software program that operates the individual nodes is installed on a Raspberry Pi (Model B, Raspberry Pi foundation, United Kingdom). Monitoring and control logic were implemented using an open platform program (FarmosV2, Jinong Inc., Gyeonggi-province, Republic of Korea) (Park et al., 2019). The control logic uses a control algorithm based on a P-band. The P-band used for ventilation is a system that determines the window opening (%) by calculating the ventilation load through the external temperature, wind direction, wind speed, and solar radiation as

the setting value through the linear coefficients. All the control signals and sensor values of the environmental controller are stored in the DB as described above and used for prediction model development and control algorithm design. d

Table 32. Sensor specifications for inside climate of the greenhouse

Component	Measurement range	Resolution	Operating temperature (℃)	Response time (Seconds)
Temperature	-10.0 ~50.0 ℃	± 0.3℃	-25.0 ~85.0	5.0 ~30.0
Humidity	0 ~99.0 % RH	± 2.0 %	-10.0 ~50.0	8.0
CO_2	0 ~ 3000 ppm	± 10.0 ~ 50.0 ppm (Proportional to measurement range)	-10.0 ~50.0	2.0

The ventilation window structure of the greenhouse is driven by a total of six windows. There are three top windows and three side windows. The top window 1 is located on the outermost side of the greenhouse and has a triangular structure that allows the wind to flow from both sides. Similarly, in the case of side window-1, it is located at the outermost side and the left and right sides open simultaneously. Operating ranges of three side window are 800 mm, 760 mm, and 650 mm, and open proportionally when the ventilation control logic determines that ventilation is necessary. All ventilation windows operate between 0 and 100% of opening and closing proportional to time. Window 1 and 3 must be opened for direct mixing with the air inside the greenhouse because of the characteristic of the experimental greenhouse, it is a 3-ply multi-window structure. In case of window 2, it can be set to determine a more minute ventilation amount, and additional insulation role is possible.

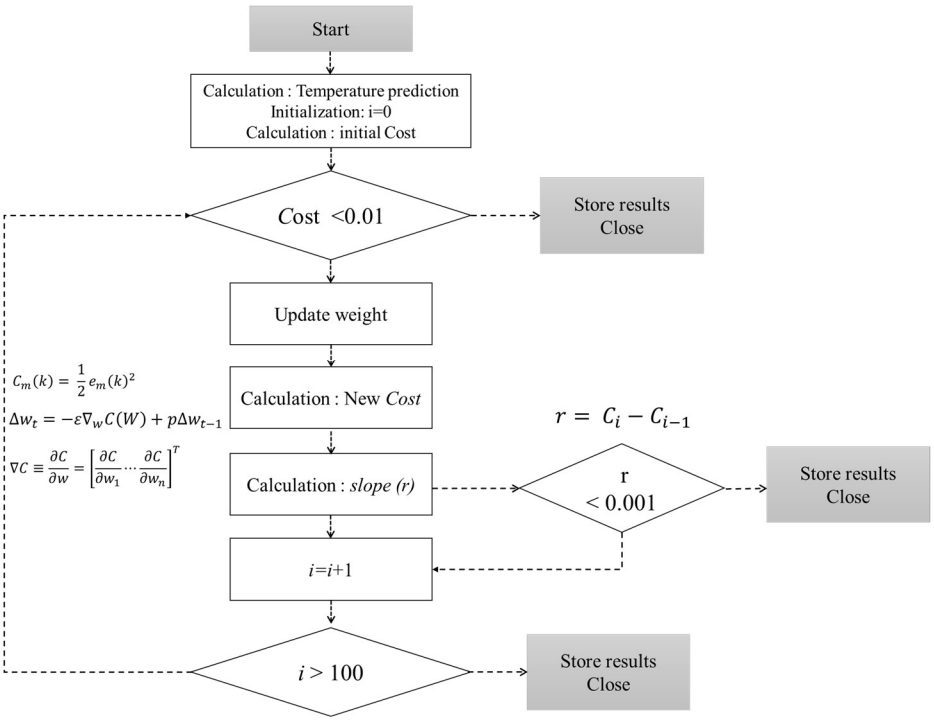

Figure 103. Flow chart of the gradient descent-based optimization.

5.3.2. SIMULATION AND FIELD EXPERIMENT TESTING

In order to test the new control algorithm, field verifications on specific conditions were conducted on May 11, 2018, using the environmental data from three time points on this day. The test was confirmed the signal decision process and decision process of the developed algorithm. At this time, the predicted temperature and cost value for each loop determined repeatedly were checked, and the decision history about multi-window opening was compared. The final result of this case simulation comparison is to figure out the operation of OFNN-based optimization algorithm in practice. In addition, the result can be confirmed by comparing the relationship between the signal change of the six window and the effect of the obtained value on the temperature prediction model. The three cases below were selected based on the three most important points of the day, after sunrise, after midday and before sunset. The three time points used are shown in the table below. The change of control signal of each window due to OFNN was observed at each time point. In each case, the external and inside environmental conditions are fixed, and the ANN and OFNN models operate repeatedly to determine the six window open signals, and the specific

environmental conditions are shown in Table 33. In addition, the ANN model and OFNN were simulated simultaneously under the assumption that the whole day was continuously controlled by the proposed method.

Table 33 Three case environmental conditions to confirm the simulation of the optimization algorithm.

	External temperature (Et)	External humidity (Eh)	Radiation (Rv)	Wind speed (Ws)	Inside temperature (It)	Inside humidity (Ih)	Inside CO_2 (Ci)	Time
Case-1	14.1 ℃	64.5 %	25W/m^2	1.4 m/s	15.2 ℃	61.2 %	454.3 ppm	06:30
Case-2	26.9 ℃	62.5 %	788 W/m^2	0.8 m/s	27.9 ℃	67.3 %	414.3 ppm	15:30
Case-3	19.5 ℃	71.5 %	105 W/m^2	2.3 m/s	21.0 ℃	55.5 %	464.6 ppm	18:30

Field tests were carried out May 18 to May 24, 2018, in which used two greenhouses (Figure 104). The controller equipped with the newly developed in this study and the controller equipped with a standard commercial controller were installed in the greenhouse, respectively, and the control performance was evaluated by setting desired temperature value. The commercial controller adopts the P-band algorithm, which is a kind of proportional parameter that determines the opening angle of the window according to an excess of desired temperature: the difference between the set point and the measured point is the reciprocal of the proportional gain constant (Kamp, 1996; Kim et al., 2017).

For comparison experiments. Both controllers used a Raspberry Pi microcontroller which received environmental sensor information and sent it to the server. In the greenhouse equipped with our control algorithm, an additional computer (Core i7-6700 processor, Intel®, USA) was installed to infer prediction models and operate subroutines for determining the control signal.

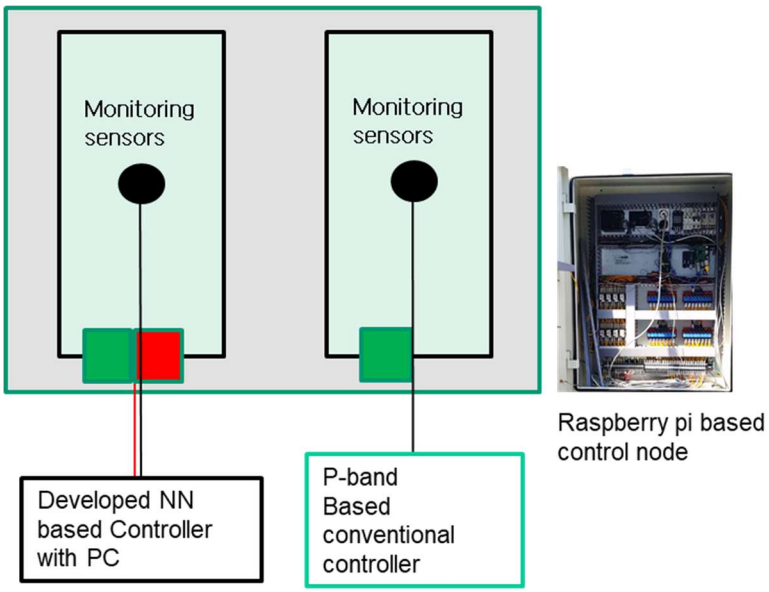

Figure 104. Comparative greenhouse and control node diagram for field application experiment.

5.3.3. PERFORMANCE OF TEMPERATURE PREDICTION MODEL

The prediction results for the ~11,000 validation samples in the learning process yielded a 0.99 R^2 for the calibration curve with a slope of 0.94, an offset of 1.53, and a total RMSE of 0.78 °C (Figure 105); this showed a very high accuracy of the developed model during training and validation. The temperature changes after 30 min ranged from -3.9 °C to 6.3 °C (Figure 106); a comparison of the predicted and measured temperature change showed an R^2 of 0.94 and an RMSE of 0.19.

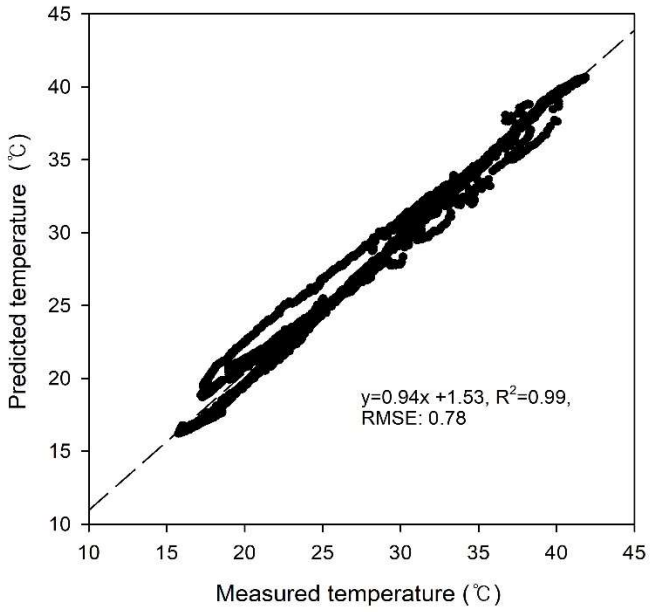

Figure 105. Comparisons between predicted and measured temperature

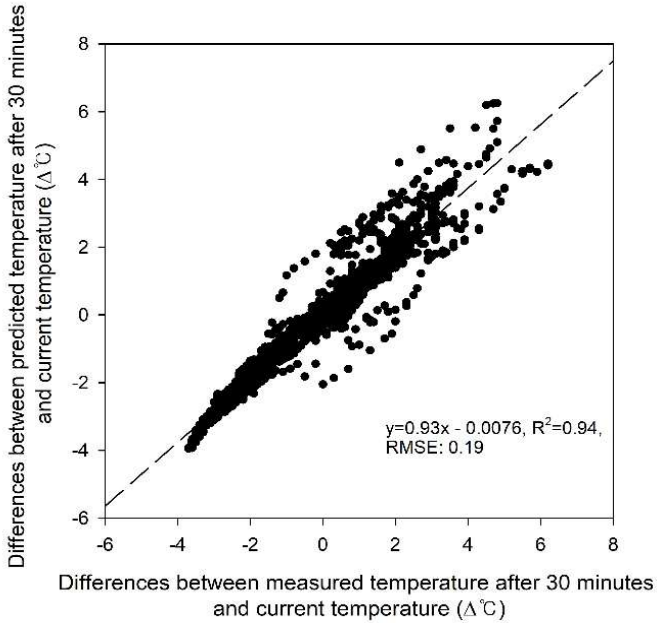

Figure 106. Comparisons between difference between current temperature and predicted or measured temperature after 30 min.

To identify the temperature prediction performance of the model, the trained model

was implemented to predict the change of indoor temperature and the predicted values were compared with measurements. In Figure 107, the temperature change in a greenhouse can be identified by the graph of about 2 days. The RMS error between the measurements and the predicted values obtained from the prediction model was about 0.91 indicating the excellent performance.

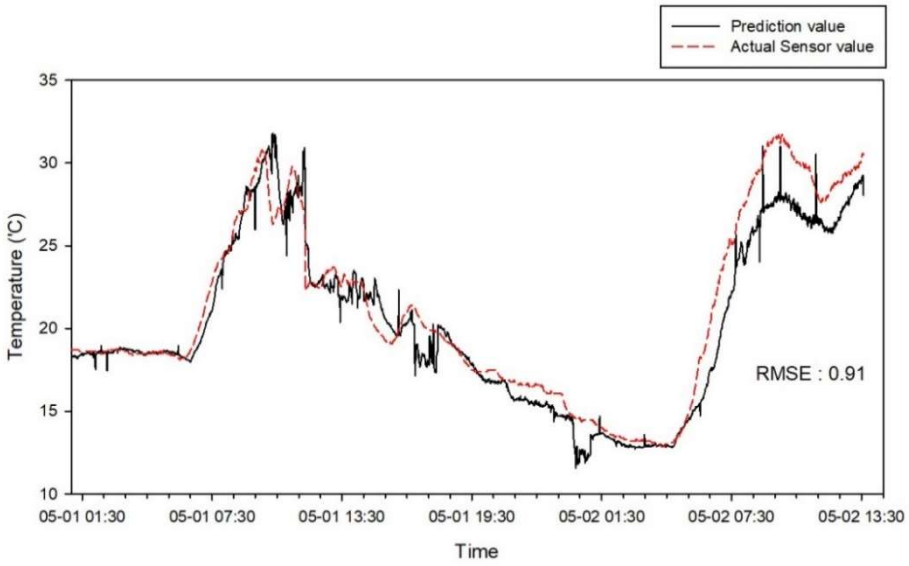

Figure 107. Performance of temperature prediction in 30 minutes.

5.3.4. SIMULATION AND FIELD TEST RESULTS

The OFNN-based control signal decision algorithm proposed in this study first performed driving verification in three cases. Figure108 shows the expected temperature change (left) and window opening change plot (right) simultaneously with the case-specific OFNN algorithm. Figure 108 (a) and (b) show the progression at 6:30, when the sun rises. The red dotted line in Figure 84 (a) is the target temperature, 15.70 °C, and the blue dotted line is the current temperature at that time, which is about 15.08 °C. The initial expected temperature is about 13.40 °C, and the temperature drop is expected below 4 epochs. In Figure 108 (b), the control algorithm initially inferred that the window was open, but since the expected temperature is far from the target temperature, the process of epoch 5-7 have given the command to close the window. The final expected temperature is 15.04 °C, and Six window openings were determined to close by 10-20%. Figure 108 (c) and (d)

show the case around 15:30, when the current temperature is about 27.91 °C and the target temperature is about 25.02 °C. In Figure 108 (d), the six windows converge on the open signal (70~90%) rather than before. Figure 108 (e) and (f) show that the inside temperature, which rises during afternoon around 18:30, suddenly drops as the sun radiation drops at sunset. The current temperature and the predicted

temperature were about 21.0 °C, in order to maintain the target temperature of 21.6 °C, the windows are converging with the closing command.

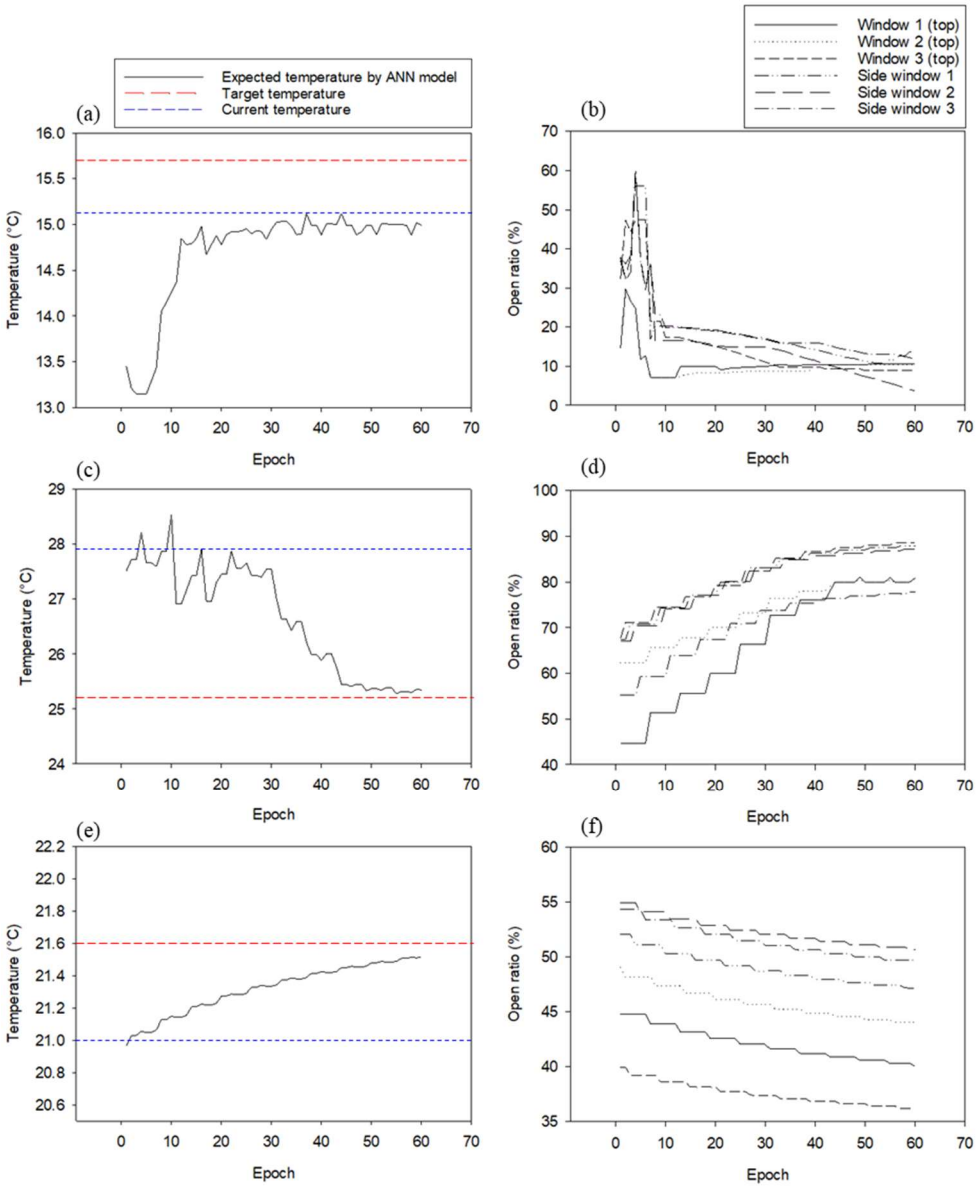

Figure 108 Three cases of predicted temperature change and window open ratio change plot as a result of the operation state of OFNN and ANN during 60 epochs; (c) expected temperature changes in case-2, (d) changes in window open ratio determined by OFNN

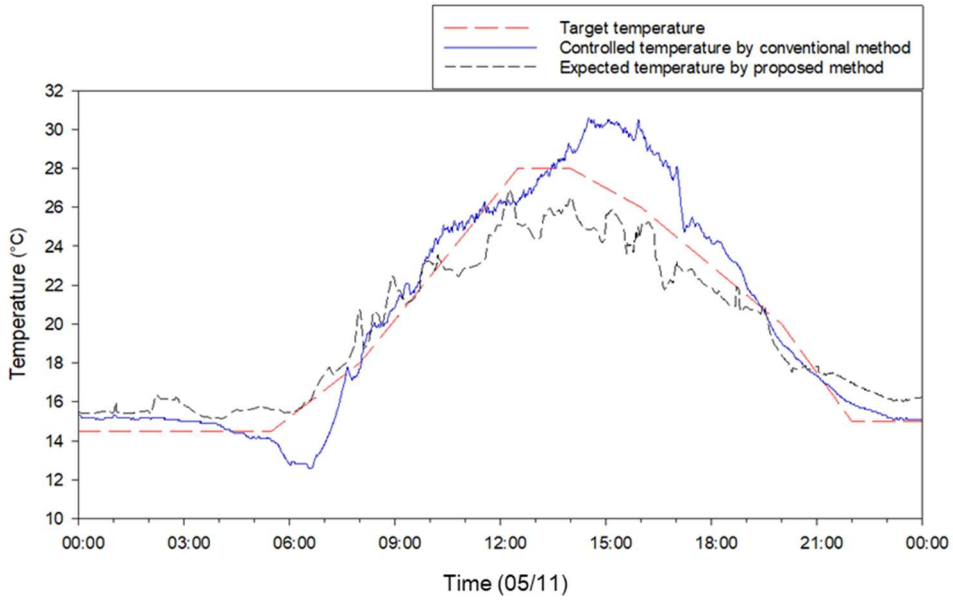

Figure 109. Comparison of temperature control performance between the proposed method and the conventional controller on 11 May. network-based controller.

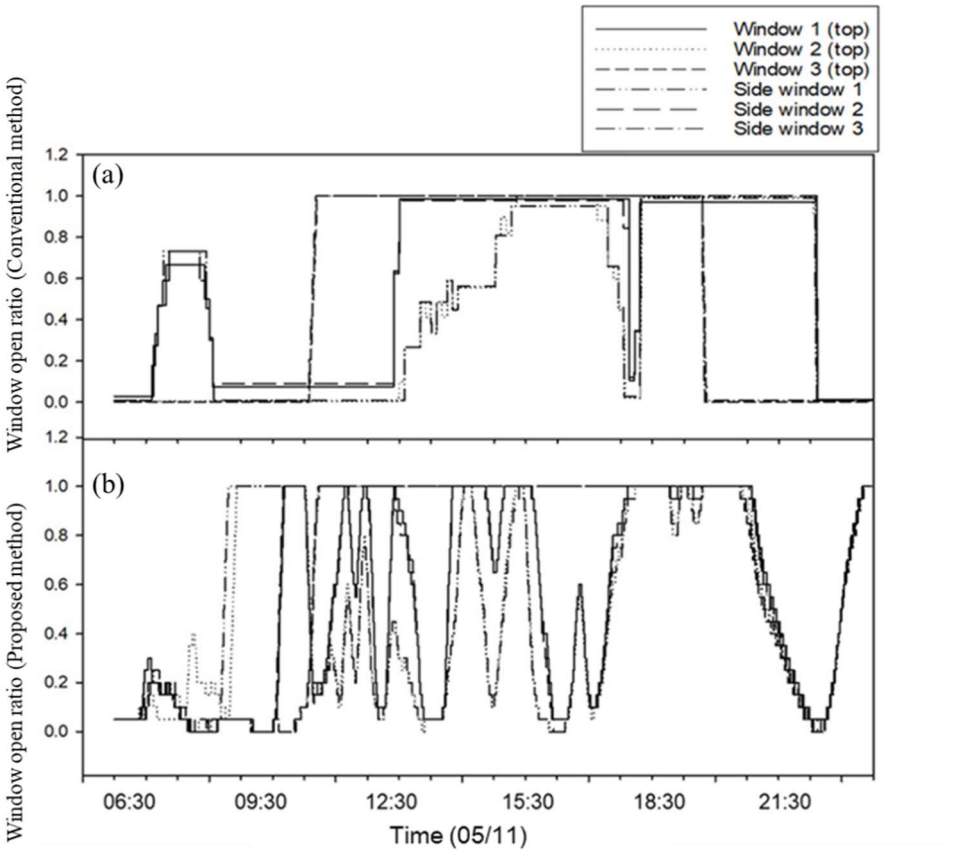

Figure 110. Comparison of opening ratios of windows between the commercial controller and the output feedback neural network-based controller under the same condition.

Finally, the OFNN based control logic was mounted on a real strawberry greenhouse, and our proposed logic and conventional ventilation control method are applied to two the same greenhouses. Over a six-day field test from 18–24 May, the RMSE of the target temperature as compared with the conventional controller (3.01 °C), higher than that for the proposed method (2.45 °C), again confirming the better control performance in the field application (Figure 110). Figure 111 and Figure 111 show changes in the environment outside the greenhouse and other environmental factors inside the greenhouse, respectively. The factors affecting the inside of the greenhouse appear to have a large proportion of outside temperature and solar radiation. In addition, it is possible to observe changes in the environmental conditions inside the greenhouse and the outside wind velocity during the field test period.

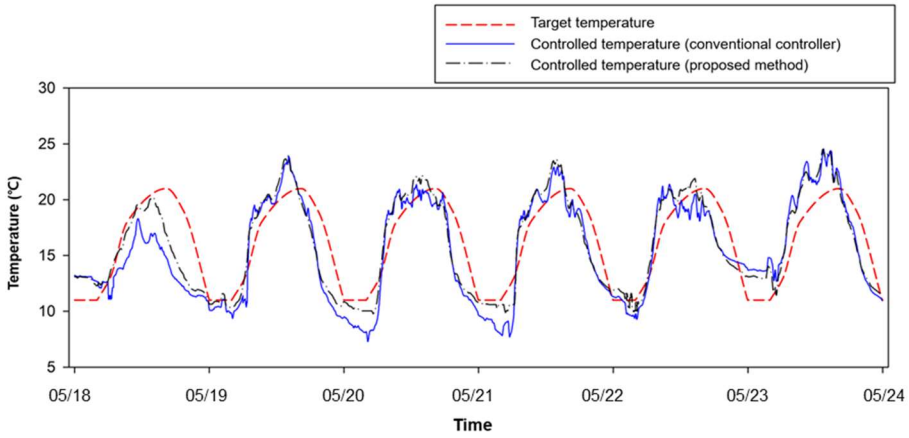

Figure 111. Field test results for the proposed method from 18 May to 24 May.

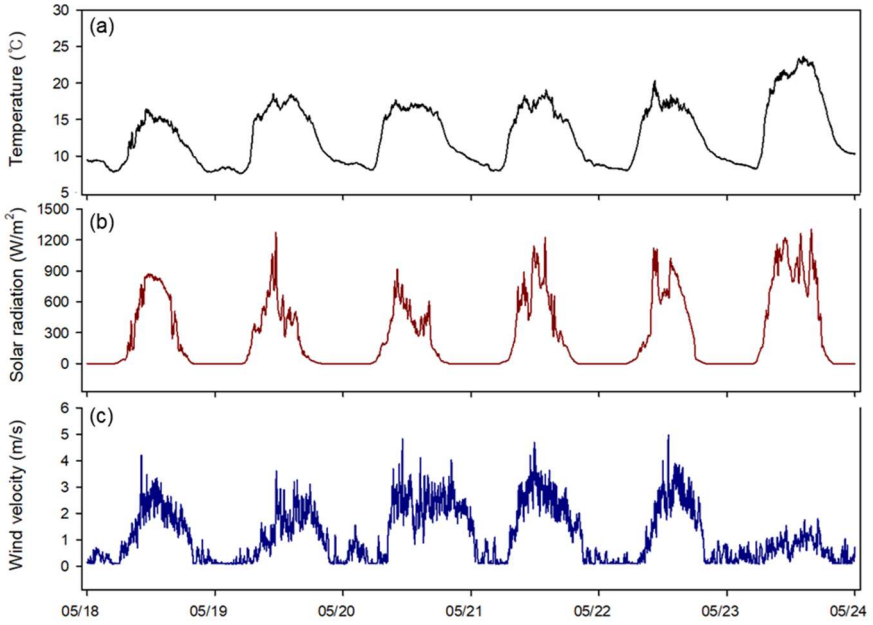

Figure 112. Changes in outside environmental conditions in the experiment: outside temperature (a), solar radiation(b), and wind velocity (c).

P-band based control logic is mainly used in commercialized greenhouse controllers (Kim et al., 2017) and requires users to input many setting values for the optimal environmental condition, which is not easy for growers. In contrast, the predictive model and control logic developed in this study were trained by greenhouse conditions without the setting values inputs separately. For example, in the 11 May test (Figure 111 and Figure 112), the actual temperature decreased sharply between 6 AM and 7 AM, caused by low-temperature air entering the greenhouse. As for the conventional control based on P-band logic, it considered much ventilation due to the influence of increasing solar radiation after sunrise and opened several windows. It is because a simple linear algorithm makes a decision to open the window in response to a sharp rise in temperature at this time. Whereas, the control method developed in this study was trained from the greenhouse data, the temperature drop due to window opening was predictable, so the controller decides to keep the window closed at this time and opened a bit (Figure 56(b)). This effect can be confirmed by the field experiments from 18−24 May.

The environmental control method based on the actual applied environmental model has been steadily proposed, and it was a major trend of applying the simulation through system identification. (Bennis et al., 2008; Shen et al., 2013). In this paper, the proposed method was implemented by the proposed optimization method, and the feasibility was confirmed through the field application results. This method is expected to be applicable not only to greenhouses ventilation control but also to ventilation management of livestock facilities or environmental management in residential building. This study evaluate that the neural network-based prediction model and control logic yields better control signals only in the greenhouse where training was conducted. In addition, the logic was applied in the external climate of spring and autumn season, and the window control was performed only for ventilation. However, humidity and radiation are also important factors affecting by ventilation. Therefore, various environmental factors and greenhouse structure should be considered by various attempts of OFNN structure and cost functions.

A sever computer was installed in the field due to overloading of the Raspberry Pi-based micro-controller. The developed logic executed up to 60 sub-routines in determining the control signal while updating the prediction model and the control

node. In Raspberry Pi, this took 1,100 ± 125ms while using 100% of the CPU, meaning that environmental monitoring and server data transfer could not be performed at the same time. In addition, the numerical values used in the optimization suggested in the study (Cost: 0.01, r: 0.001) could find the optimal convergence conditions in different ranges for other applications, and it is possible to compute the numerical analysis faster by adjusting these values. Solving this problem could be achieved through parallel algorithm optimization or cloud computing technology that implements real-time control algorithms.

5.4. CHAPTER CONCLUSION

In this study, two techniques were tested for effective multi-window ventilation control in greenhouses. First, a temperature prediction model based on neural networks was developed for the monitoring of the internal environment of a greenhouse. Many studies based on predictive models have reported positive results for machine-learning or deep-learning technology in the modeling of the greenhouse environment (Baumeister et al., 2017; Andreas Kamilaris and Prenafeta-Boldú, 2018; Maher et al., 2016). As shown in Figure 105, the model developed to predict changes in temperature performed well (R^2: 0.99; RMSE: 0.78), showing that model-based control was feasible due to the precise predictive model. Prediction models for various time periods should be developed in the future in order to improve control performance.

The second technique was the use of control signal decision algorithms to optimize the OFNN structure. The nonlinear relationship between the change in temperature and the ventilation rate of six windows was overcome by using the SGD with momentum method. The proposed OFNN-based control logic was implemented in a real strawberry greenhouse and compared with a conventional ventilation control method under the same conditions. Over a six-day field test from May 18−24, the RMSE for the target temperature using the conventional controller (3.01 ℃) was higher than that for the proposed method (2.45 ℃), again confirming its more accurate control performance in the field. It can be concluded that the proposed method provides a promising foundation for field applications in greenhouse environmental modeling research.

6. DEVELOPMENT OF ARTIFICIAL INTELLIGENCE CONTROL LOGIC CONSIDERING THE ENERGY USE EFFICIENCY OF GREENHOUSE

Sustainable horticulture is an important goal in the agricultural industry because the global increase in the population has led to the need for higher production yields, which in turn has led to a rise in the energy demands of the industry (Vadiee and Martin, 2012). Because energy use accounts for the majority of greenhouse production costs, reducing energy costs is an important goal for the greenhouse industry (Mohsenipour et al., 2020). The increase in productivity per unit of energy is possible by reducing the energy use or improving the energy efficiency. Although most cultivation methods that increase production are based on greater energy inputs, setting up a control scheme without considering the energy consumption can have serious negative consequences for the environment.

In general, the purpose of greenhouses in East Asia is to maximize the amount of solar radiation energy in autumn and winter, while minimizing energy losses and reducing the high temperatures in spring and summer. It occurs even though the lowest electricity price rate is set for energy facilities such as heat pumps and electrical boilers in the agricultural sector in South Korea. The relatively low cost for electricity may in fact lead to the careless waste of energy, but this would also suggest that there exists significant opportunity to reduce energy consumption by adopting energy-efficient measures that are commonly used in other sectors, such as in residential buildings or the manufacturing industry (Lee et al., 2019).

The first step in improving the energy efficiency of a greenhouse is to maximize the use of solar radiation and minimize fuel consumption for heating. The next step is to use natural ventilation to control convection, which is the main source of energy loss, radiant energy emitted from the greenhouse surface, and latent heat. Therefore, the opening and closing of the windows and the optimal use of the heat retention and shade curtains are very important. Finally, in order to maximize the benefits of greenhouse environment management, a management system that considers energy

consumption in terms of the external climatic conditions is required, rather than continuously operating the controls for heating, ventilation, and CO_2 supply.

There has been an increase in the use of energy in the agricultural sector arising from greater access to machinery and facilities (Yu et al., 2016b). However, this has increased input costs such as heating costs and worsened greenhouse gas emissions. This chapter outlined the status of energy use in the facility-based horticulture sector, examined the changes in duty-free oil and electricity for farming, and controlled the supply status energy saving facilities for agriculture (Benni et al., 2016a). Currently, agricultural operations have to adapt to a more competitive environment and consequently use new intelligent technologies (Shamshiri et al., 2020).

The purpose of this chapter is to design variable optimization control for energy consumption by adding a cost gate modified by the OFNN backpropagation method to climate control within a greenhouse using AI. The energy-saving effect of this design in a smart greenhouse is then tested.

6.1. DESIGN OF ENERGY USE EFFICIENCY IN COST FUNCTION

6.1.1. ENERGY CONSUMPTION MODULE

Equation [56] is a partial differential optimization function, where θ is the parameter, *m* is the number of samples, and *x* and *y* are the predicted and target values. α is an index for the amount of heat consumed by the control actuator in controlling the temperature, β is a coefficient for humidity, and γ is a coefficient for CO_2. These values are updated to reflect the calculated values, thus excluding the initial inputs. The updated values were calculated using $\alpha = 0.1 \cdot CS_{IT}/5.0$ and $\beta = 0.1 \cdot CH_{IH} / 20$, $\gamma\theta = 0.05 \cdot CC_{IC}/5.0$ and applied to the newly updated node. CS_{IT} is the sum of the energy consumption of the control actuator that affects the temperature inside the greenhouse, and CH_{IH} and CC_{IC} correspond to the humidity and CO_2, respectively. The active function of the OFNN employed the sigmoid function, and the conditions of the input nutrient solution were determined using two nodes. The optimization method employed momentum-based SGD, shown in Equations [9] and [10]. The number of repetitions was fixed at 50, and the final determined temperature,

humidity, and CO_2 concentration were compared with the target values, which were applied at the time of determination for control determining. Optimization logic was applied to the test set data to derive the simulation results.

$$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{1}{2m}\ [\sum_{i=1}^{m} h_\theta T_i\left(x_t{}^{(i)} - y_t{}^{(i)}\right)^2 + h_\theta H_i\left(x_h{}^{(i)} - y_h{}^{(i)}\right)^2 + h_\theta C_i\left(x_c{}^{(i)} - y_c{}^{(i)}\right)^2 + Ec\theta] \quad \cdots [56]$$

Where,
T_i : Temperature cost compensation value (1.0 used)
H_i : Humidity cost compensation value (0.125 used)
C_i : CO_2 cost compensation value (0.025 used)
E_c : Energy consumption function ($\alpha + \beta + \gamma$)

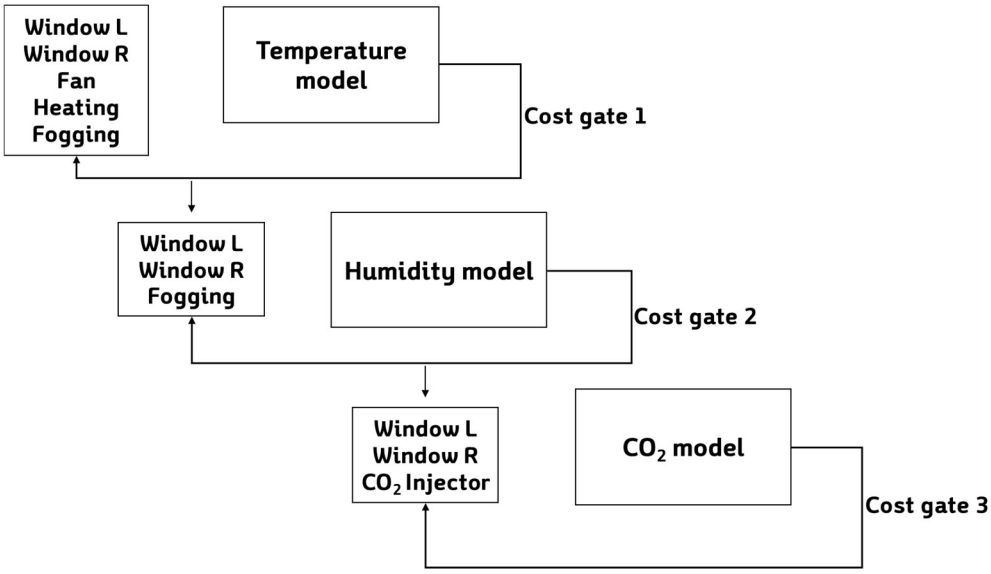

Figure 113. Cost gate design for climate control signal determination in a smart greenhouse.

6.2. SIMULATION STUDY CONSIDERING ENERGY OPTIMIZATION IN DEEP LEARNING MODEL AND OFNN STRUCTURE

6.2.1. GREENHOUSE OPERATING COSTS FOR ACTUATORS

In this study, a method for optimizing the signal of the controller using a predictive model was developed. The energy cost gate was designed so that it would be possible to build a cost map by applying the actuator's energy consumption rate as a function. The purpose of a cost function that considers energy efficiency was to find an optimal

combination by comparing the amount of energy consumed by the actuators in the greenhouse. The required energy consumption for the actuators is summarized in Table 34. This table is applicable only to the actuators in the demonstration greenhouse described above. The energy consumption was calculated based on three assumptions:

1) The amount of power consumed depends on the product specifications and does not take into account changes in efficiency due to the aging of the equipment.
2) The power consumption rate for actuators is linear.
3) The operating cost of the actuators is calculated in minutes based on the values determined by the control signal.

Energy consumption was compared in watts because the actuators used electrical energy. For the kerosene boilers, the equation 1 kWh = 860 kcal was used to convert 8,770 kcal per liter of kerosene into watts. The water heated by kerosene was expressed as an energy function for heating 1 ton of water based on the specific heat capacity of water.

$$Q1\ \ (kJ) = (T_t - T_c) * C_v * 1{,}000\ \text{kg}\ \ (\text{volume}) \quad \ldots [57]$$

$$Q2\ \ (\frac{kj}{h}) = H_{h,kerosene}\ \ * B_{C,rate}\ \ * B_{efficiency} \quad \ldots [58]$$

Duration (h) = Q1 /Q2

λ_k: Burnout time per liter for the kerosene boiler (6.78 L/h)

C_v: Specific heat for a constant volume of water (4.185 kJ/kg K)

T_c: Current water temperature

T_t: Target water temperature

$H_{h,\ kerosene}$: High heating value of kerosene (43,400 kJ/kg)

$B_{C,rate}$: Fuel consumption rate of the boiler (13.57 kg/h)

$B_{efficiency}$: Rated thermal efficiency (91.3%)

Table 34. Energy consumption table for the actuators in the experiment greenhouse (agricultural electricity rates: 49.09 KRW/kWh; CO_2 price: 300 KRW/kg; kerosene: 1,000 KRW/L; water: 360 KRW /ton)

	Actuators	**Operation type**	**Energy consumption**	**No./area**	**KRW/h**
Ventilation	Windows	Electric motor	0.25 kWh	4	49.09
Fogging	Spraying nozzle, pump	On/off electric valve Electric motor Water	0.07 kWh, 0.35kWh, 6L/h	1	20.6178
CO_2	Spraying nozzle	On/off electric valve CO_2 Gas	0.12 kWh 0.19 kg/h	1	61.909
Heating - boiler	Kerosene, pump	Fuel consumption rate, electric motor	13.57 L/h, 0.65kWh		6811.91
Heat retention curtain		Electric motor	0.25 kWh	2	24.54
Shade curtain		Electric motor	0.25 kWh	2	24.54
Circulator	Fan	Electric motor	0.036 kWh	6	10.60

To simulate the control signal roadmap, including the energy cost gate, the OFNN model was run for a current temperature of 26.4 °C and a humidity of 35.5% at 1:00 PM (Figure 114) in a March climate. The target internal environment temperature was 25.0 °C, and the humidity was 50%. Fogging was dictated by an on/off controller, but a linear algorithm from 0%to 100% was used in which, if the outcome was 50 or more, fogging took place. In the linear logic of the P-band-based algorithm, the fogging command and the ventilation signal were set at 72% open. This was because the external solar radiation was intense, and the temperature was continuously rising. In this situation, when an OFNN cost map with the energy consumption rate was constructed (Figure 114), the lowest cost occurred at a fogging index of 100% and a

ventilation index of 14.5% or 37%. This means that the loss in heat due to fogging was considered through learning, and the cost function moved in the direction of less ventilation to increase the efficiency of the fogging because its energy consumption was higher than that of ventilation.

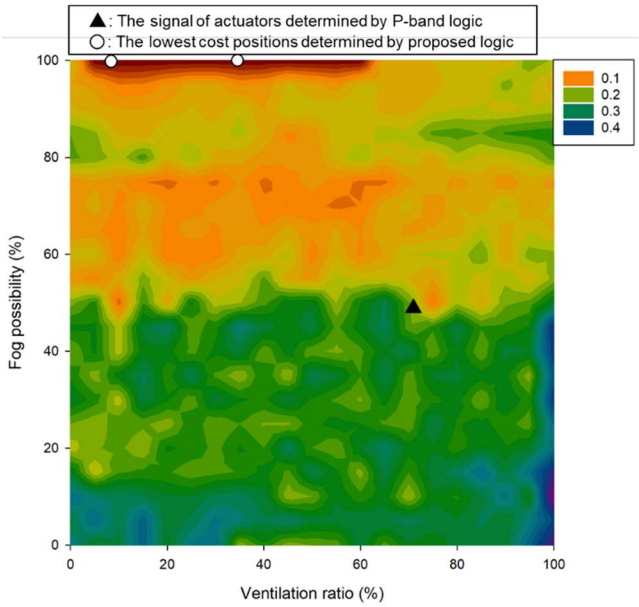

Figure 114. Cost map with energy consumption rate for determining control signals for fogging and ventilation.

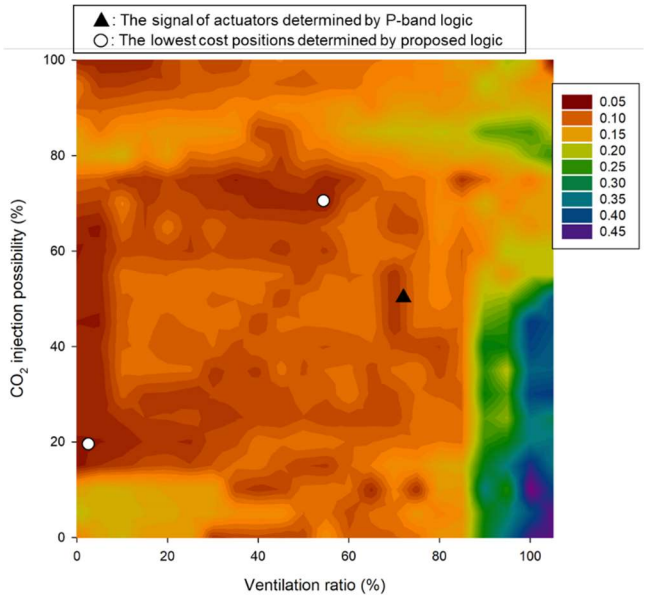

Figure 115. Cost map with energy consumption rate for determining control signals for CO_2 injection and ventilation.

The performance of this control logic was directly compared. The signals for temperature control and the actuators related to temperature control. Figure 115 presents the signal and temperature as determined based on P-band control. In P-band logic, window control is frequently used to set the target temperature for ventilation, and this causes a change in the fogging control signal. This can be confirmed in Figure 100. The RMSE for temperature based on the P-band logic was 1.78, and the most extreme change in temperature was Δ0.48 every 30 minutes on average. Figure 116 presents the results of applying the AI cost-saving logic, which leads to an RMSE of 1.65 and a temperature change of Δ0.41 every 30 minutes. This confirmed that the AI control logic was more effective than the linear control.

For humidity control, RMSE 8.5% in AI-based logic in P-band-based control obtained 10.4%, Δ 3.21% per 30 min in linear control, and Δ 5.32% per 30 min in linear control in AI control. Linear control was also more effective for CO_2 control (Figure 119), which appears to be a signal decision to make a more favorable decision for energy consumption and temperature control.

Figure 120 presents the results when the energy consumption rate was applied. The

energy consumed by the P-band logic was 8,234 KRW per day, compared to 6,000 KRW per day for the AI logic, representing an energy-saving effect of 27.10%. More than 90% of the energy use was for kerosene consumption in the control of the hot water supply. If cost control is pursued in an experimental environment where independent control is possible, greater cost savings could be obtained. However, the results of this experiment suggest that, if costs are given too much weight in energy consumption calculations, climate control may become less accurate.

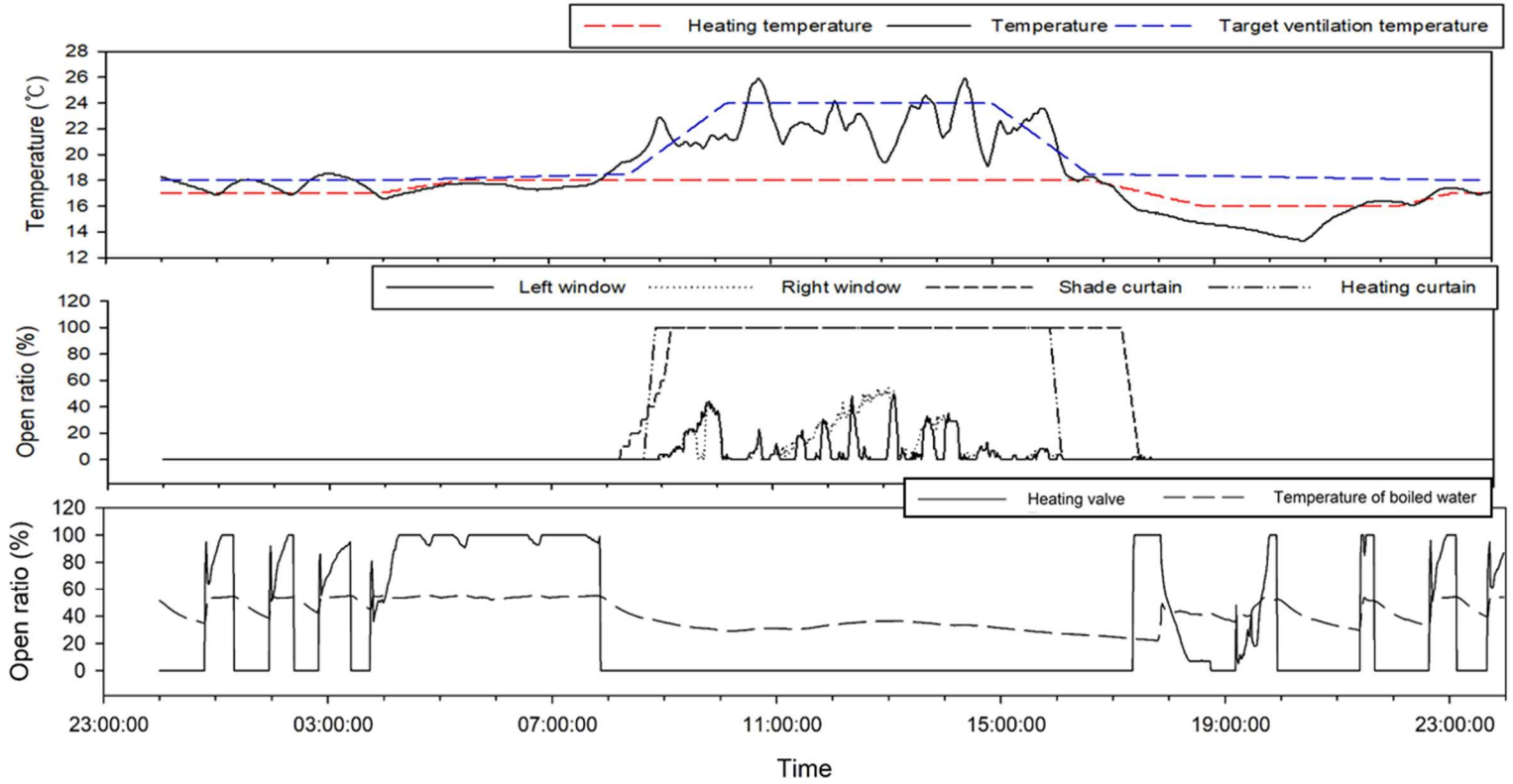

Figure 116. Change in the temperature (top), window and curtain signals (middle), and on/off actuator signals (bottom) for the P-band logic-based climate control system.

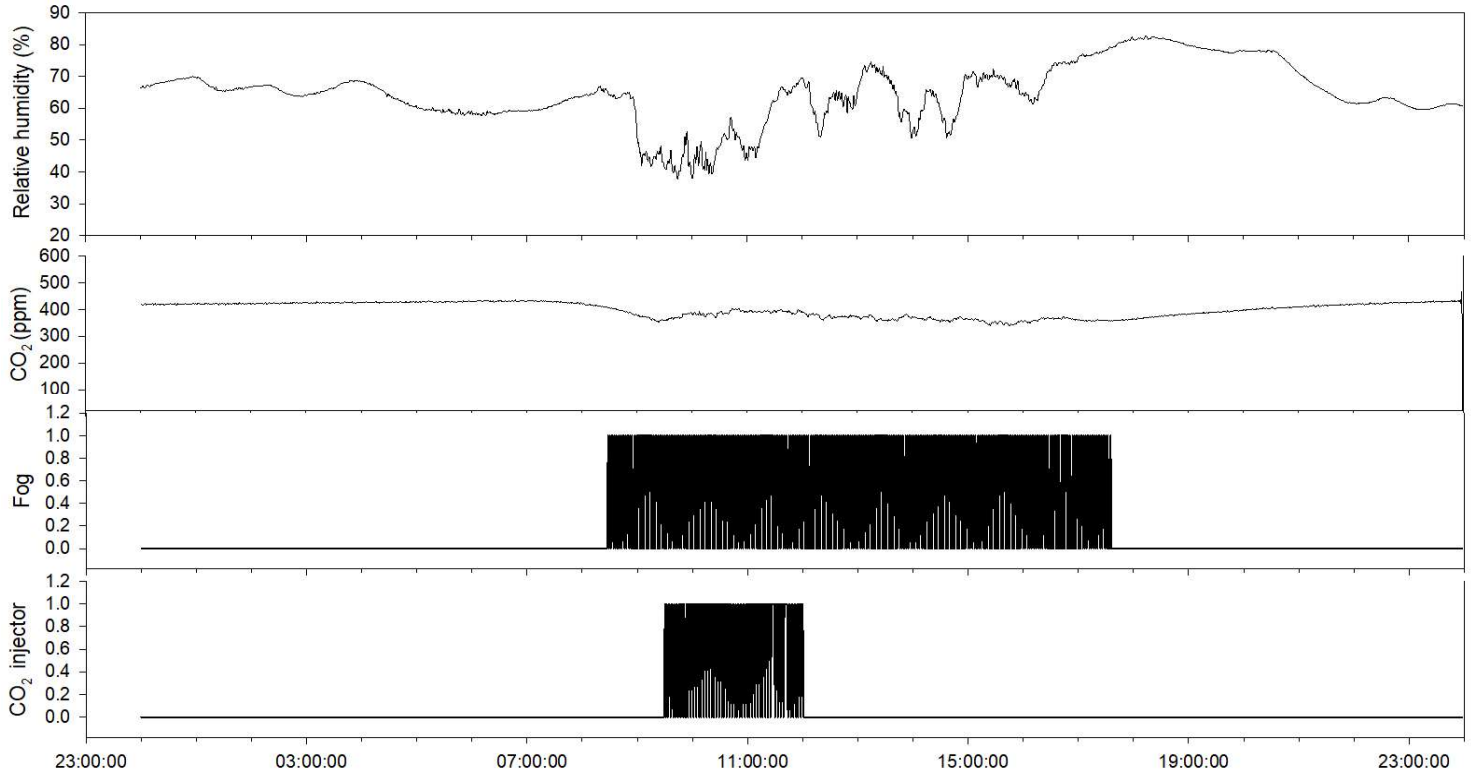

Figure 117. Change in the relative humidity, CO_2, the on/off fogging signal, and the CO_2 injector signal for the P-band logic-based climate control system.

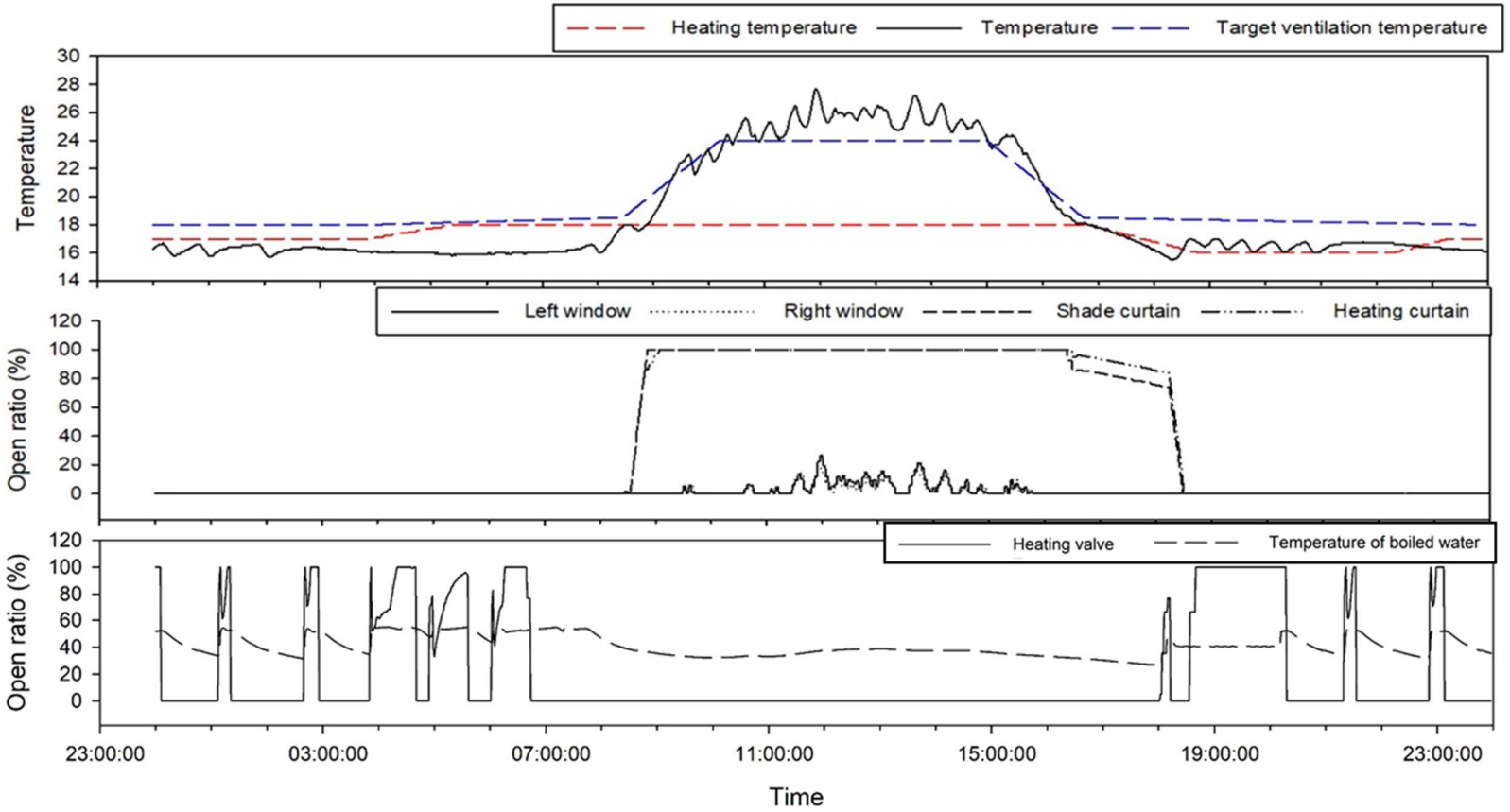

Figure 118. Change in the relative humidity, CO_2, on/off fogging signal, and CO_2 injector signal for the AI logic-based climate control system.

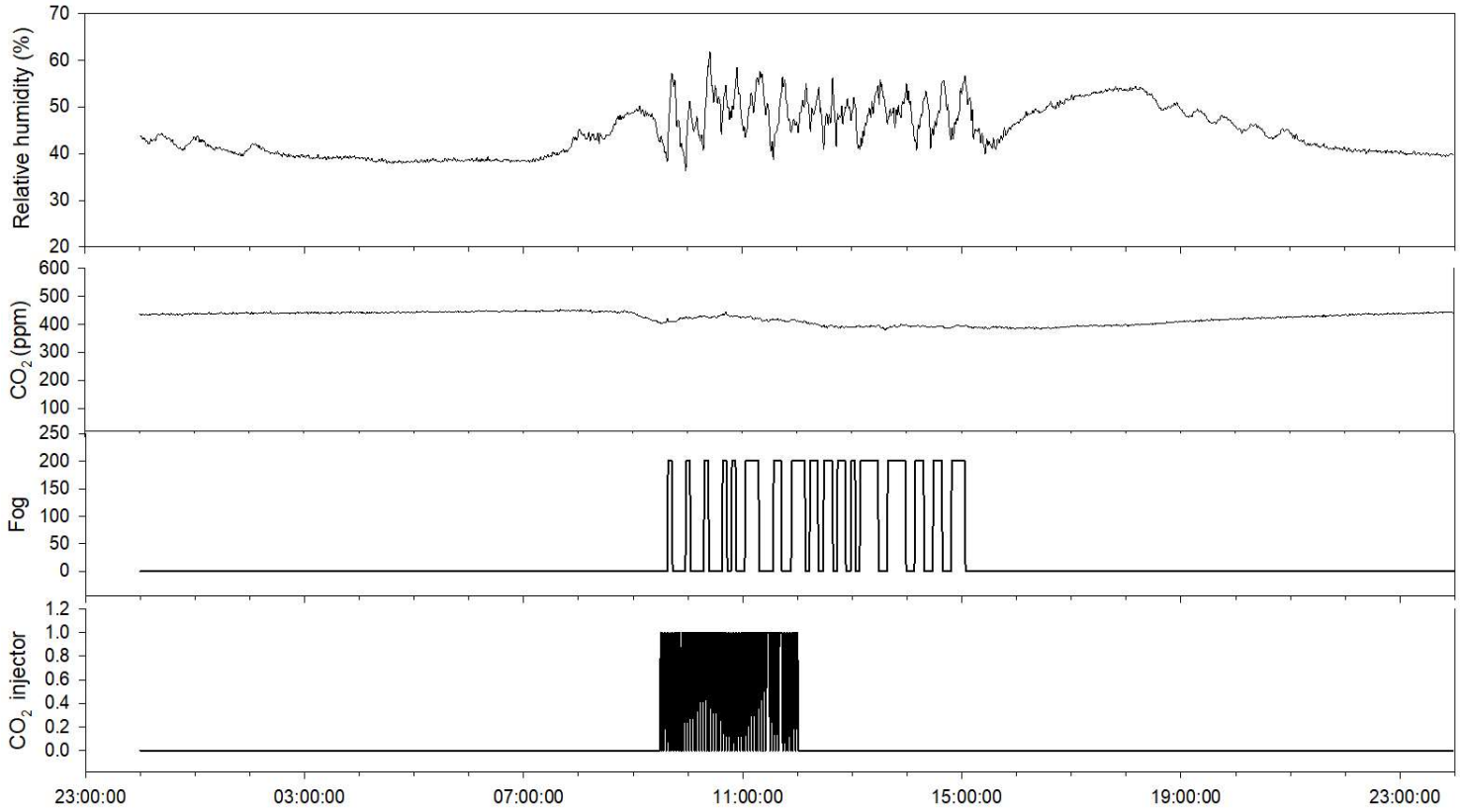

Figure 119. Change in the relative humidity, CO_2, on/off fogging signal, and CO_2 injector signal for the AI logic-based climate control system.

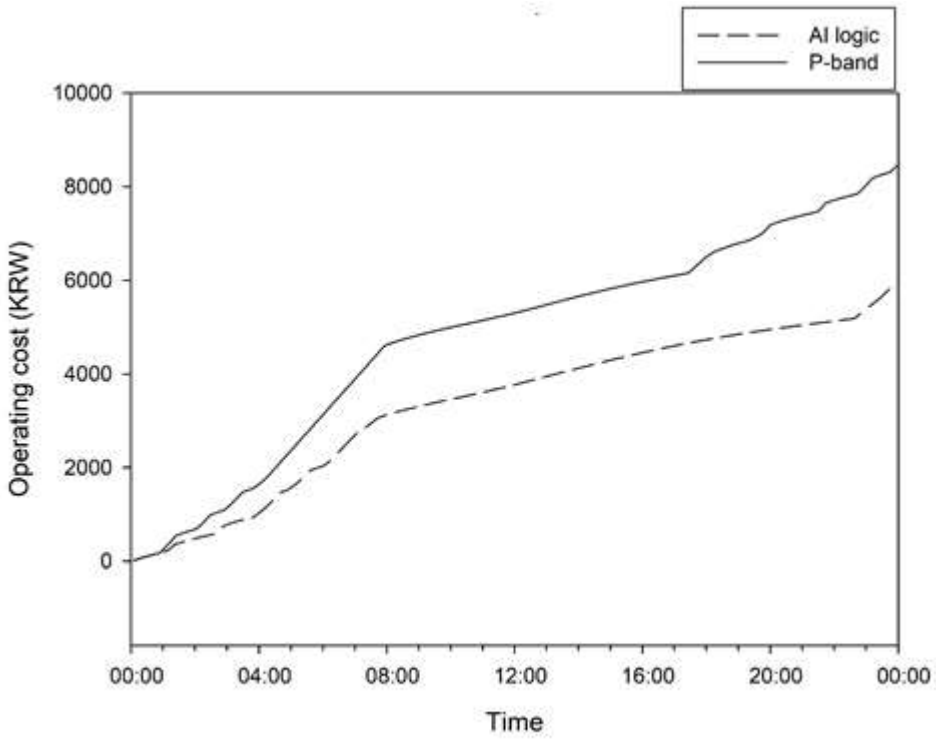

Figure 120. Comparison of the energy consumption for the linear algorithm (P-band) and AI logic control systems.

6.3. AI CONTROL SIMULATION ANALYSIS RESULTS BY SEASONAL CLIMATE

The AI-based logic system designed to account for energy consumption was subsequently tested via simulation using real data from three distinct seasonal periods. The aim was to determine whether there were any differences between the AI-based model performance and the actual results for the linear control of the greenhouse during these periods. The AI logic considering the energy consumption performed by the simulation was performed in the form of a simulation to control the actual data and intensive analysis. This reflects the driving characteristics of the seasonal actuators, and what is different from the actual controlled results, and research was conducted to confirm in advance the abnormal behavior when the AI algorithm is applied.

Three consecutive days of data were used for each season. The temperature and climatic characteristics for each period are presented in Table 35. The AI logic control system was the same as that described in Chapter 6.1, and the target values used for the cost function were those set for the actual control logic.

Table 35. Climate conditions for the three seasonal periods used in the simulation.

	Experimental days	Average outside temperature (℃)	Average outside relative humidity (%)	Average outside CO_2 concentration (ppm)
Winter season (cold climate)	Jan. 1–3, 2020	-0.54 ± 5.22	43.25 ± 3.22	422.33 ± 15.21
Spring season (warm climate)	Mar. 16–18, 2020	18.25 ± 6.27	52.44 ± 5.62	420.85 ± 21.03
Summer season (hot climate)	Jun. 1–3, 2020	24.56 ± 4.11	62.61 ± 7.11	410.66 ± 12.51

6.3.1. COMPARISON OF SIMULATION RESULTS IN WINTER CLIMATE

The performance of the proposed method in a winter climate was verified using a simulation of three consecutive days. Figure 123 presents the temperature, humidity, and CO_2 concentration for actual P-band control and for simulations with the proposed model. For temperature, the RMSE for the proposed model (2.01) was higher than that for the actual P-band control system (1.85). For humidity control, the two methods did not deviate significantly from the specified humidity range while, for CO_2 control, the RMSE for the actual control system was 57.22 ppm, compared to the simulated RMSE of 34.55 ppm for the AI-based control method. The change in the CO_2 levels for the actual control system was unusual, especially the unexpected increase in the CO_2 concentration around 9:00 PM on the first day. It is possible that this was due to an equipment malfunction, such as a leak from the solenoid valve. This is the main reason why the simulated AI-based approach exhibited better performance.

When comparing the energy consumption patterns, it was found that the AI logic tended to limit fan operation while using fogging when the P-band logic control did not. In general, in winter, it is often not necessary to use the fogging device because it is more important to maintain the temperature inside the greenhouse. In this logic, the internal humidity and temperature are affected by fogging (Figure 121). As observed in Figure 122, energy costs of 81,182 KRW were incurred by the actual P-band controlled greenhouse, compared to only 72,579 KRW for the simulated AI logic control, a reduction of 10.59%. Most of this reduction was due to the heating boiler, which is thought to have resulted in the weaker temperature control performance.

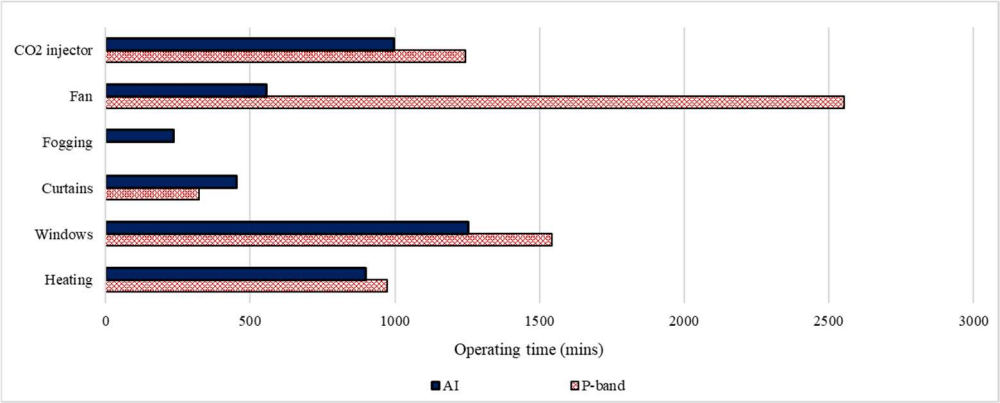

Figure 121. Operating times for the actuators over three days in winter.

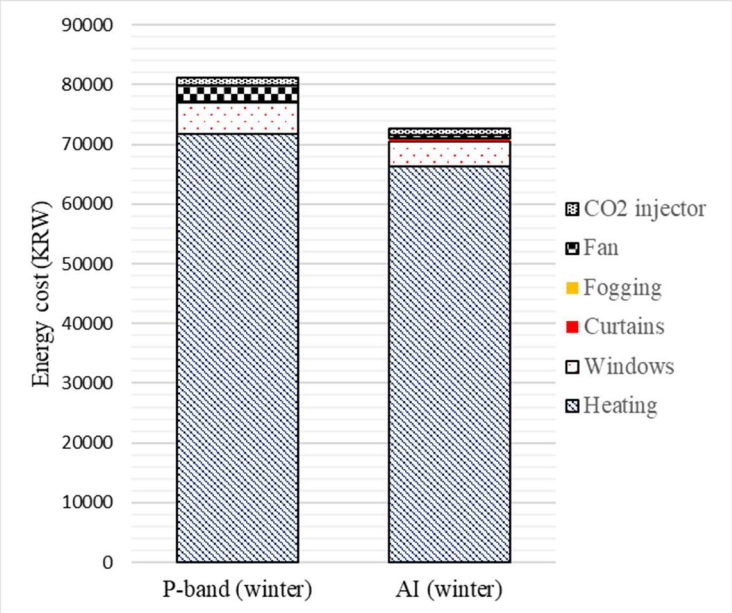

Figure 122. Operating costs for the actuator over three days in winter.

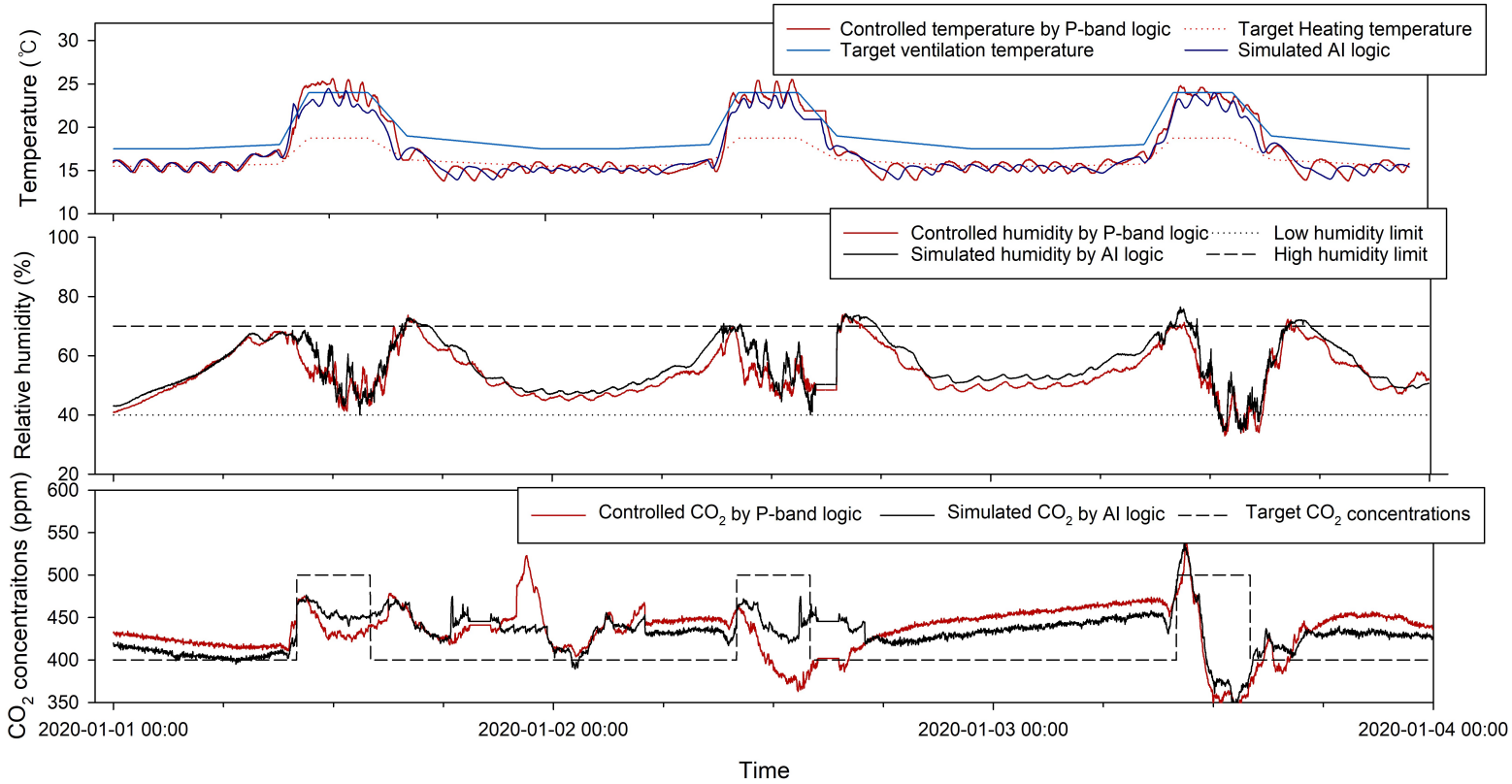

Figure 123. Comparison of simulated control performance with AI logic and actual P-band control performance for temperature, humidity, and CO_2 levels in winter.

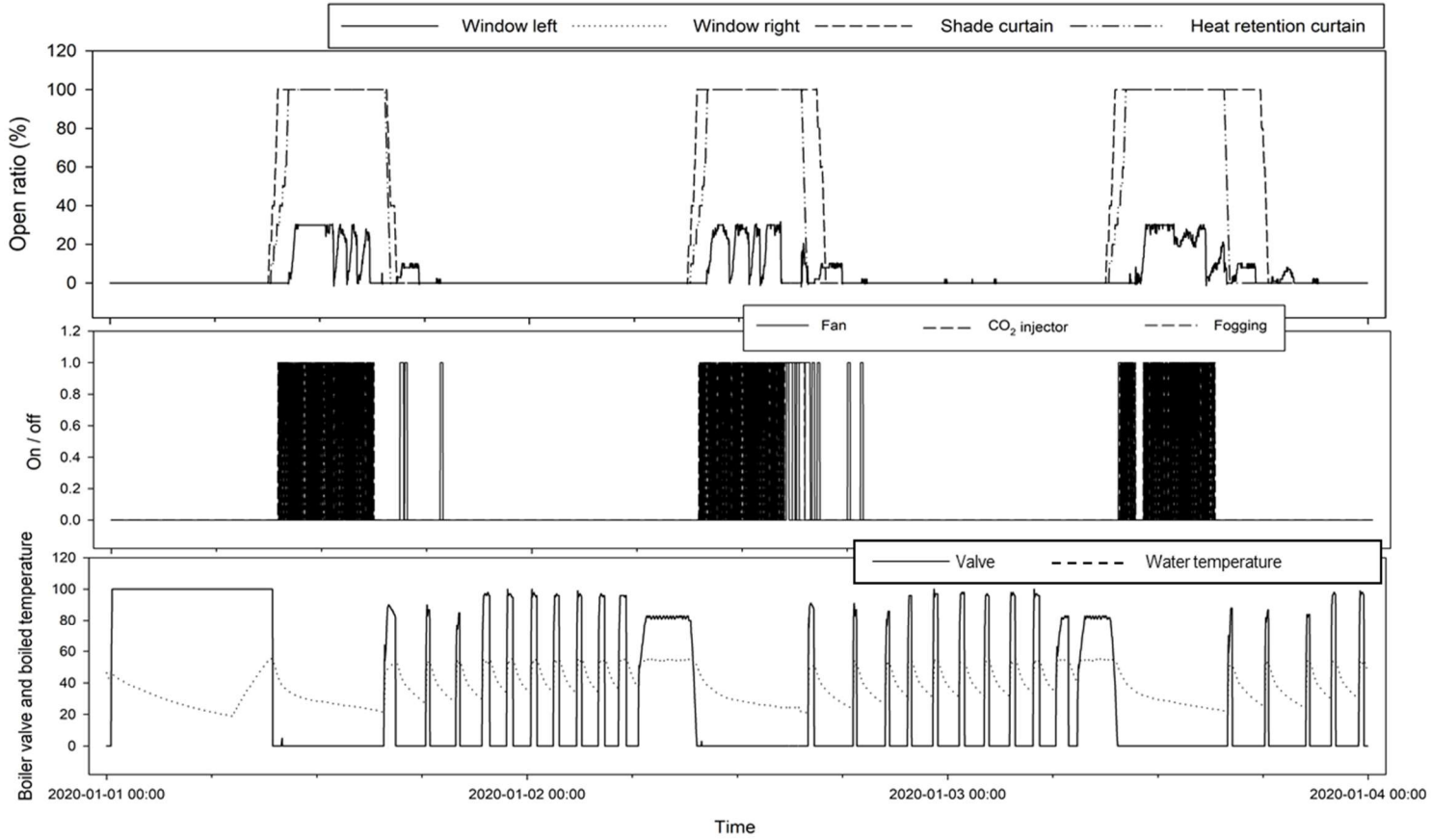

Figure 124. Control history for the actual P-band-controlled actuators inside the greenhouse in winter.

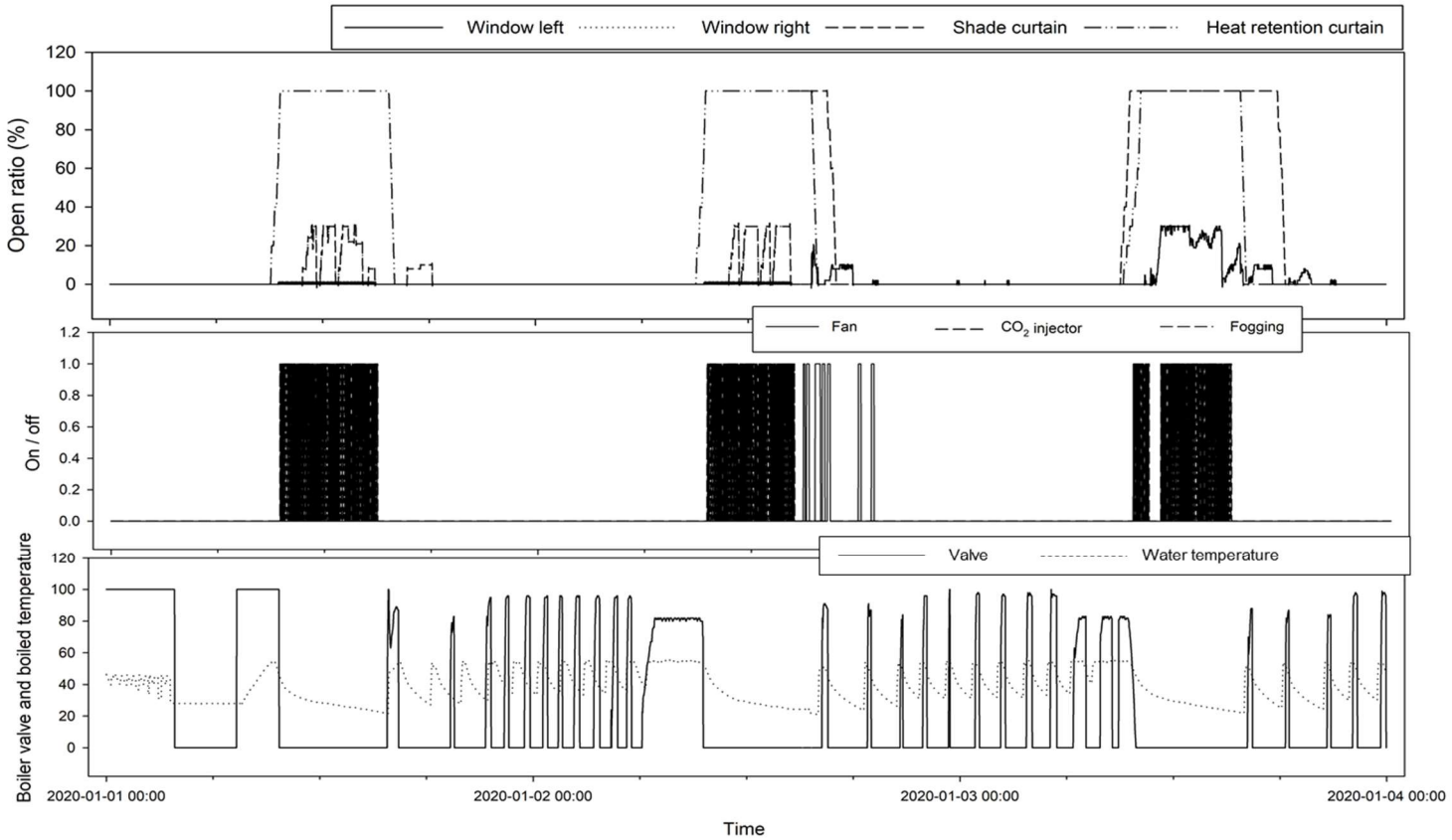

Figure 125. Control history for the simulated AI logic-controlled actuators inside the greenhouse in winter.

6.3.2. COMPARISON OF SIMULATION RESULTS IN SPRING CLIMATE

The performance of the proposed AI-based control system in spring was assessed using a simulation of three consecutive days. Figure 123 presents the temperature, humidity, and CO_2 concentration for the actual PD-band logic control system and those simulated for the proposed model. In terms of temperature control, the simulated AI logic (RMSE: 1.94) outperformed the actual PD-band control system (RMSE: 2.23). Similarly, the proposed method was more accurate for humidity (RMSE of 3.45 and 4.22, respectively). For CO_2 levels, the RMSE for the actual control system was 35.01 ppm, while the simulated RMSE for the AI-based control system was 32.11 ppm.

When comparing the energy consumption patterns, it can be seen that, compared to winter, the use of the boilers was significantly lower, while the opening of the windows for ventilation increased. The AI logic made the decision to limit the operation of the fans and was slightly more active in terms of ventilation than the real actuators were (Figure 126). Figure 127 shows that 56,367 KRW was required to power the PD-band controlled greenhouse, while 45,216 KRW was required for the simulated AI-based control system, a reduction of approximately 19.78%. This reduction was primarily achieved by limiting the use of the boiler and the ventilation windows.

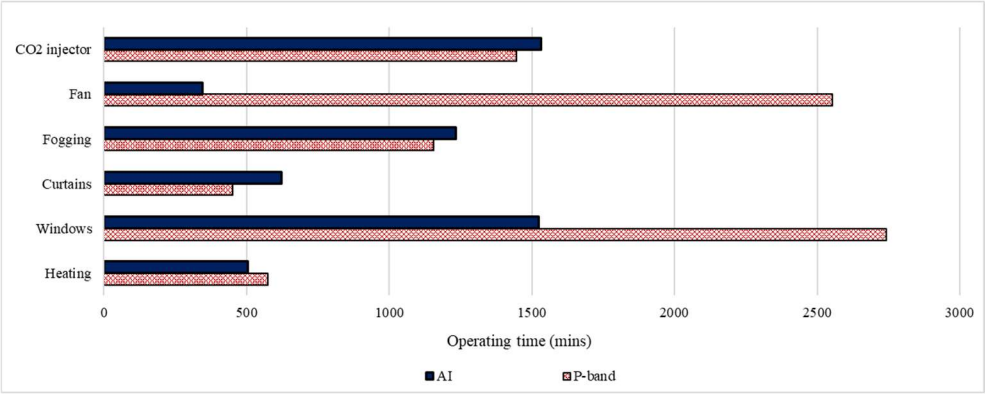

Figure 126. Operating times for the actuators over three days in spring.

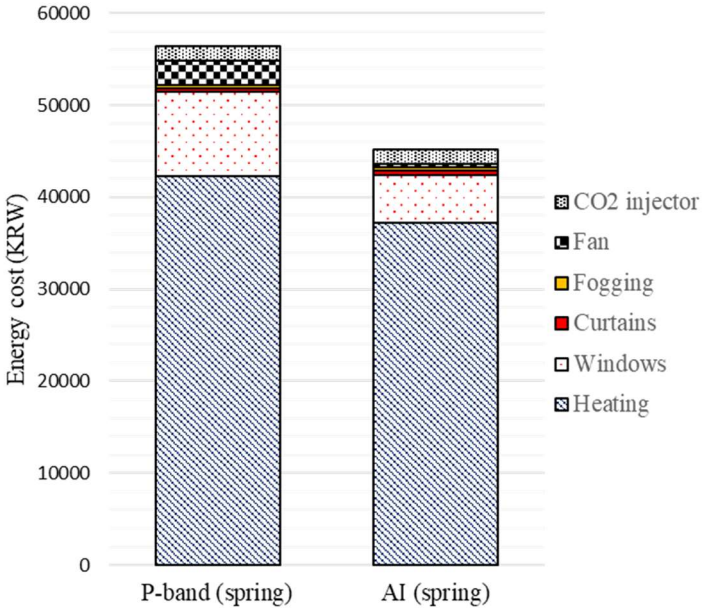

Figure 127. Operating costs for the actuators over three days in spring.

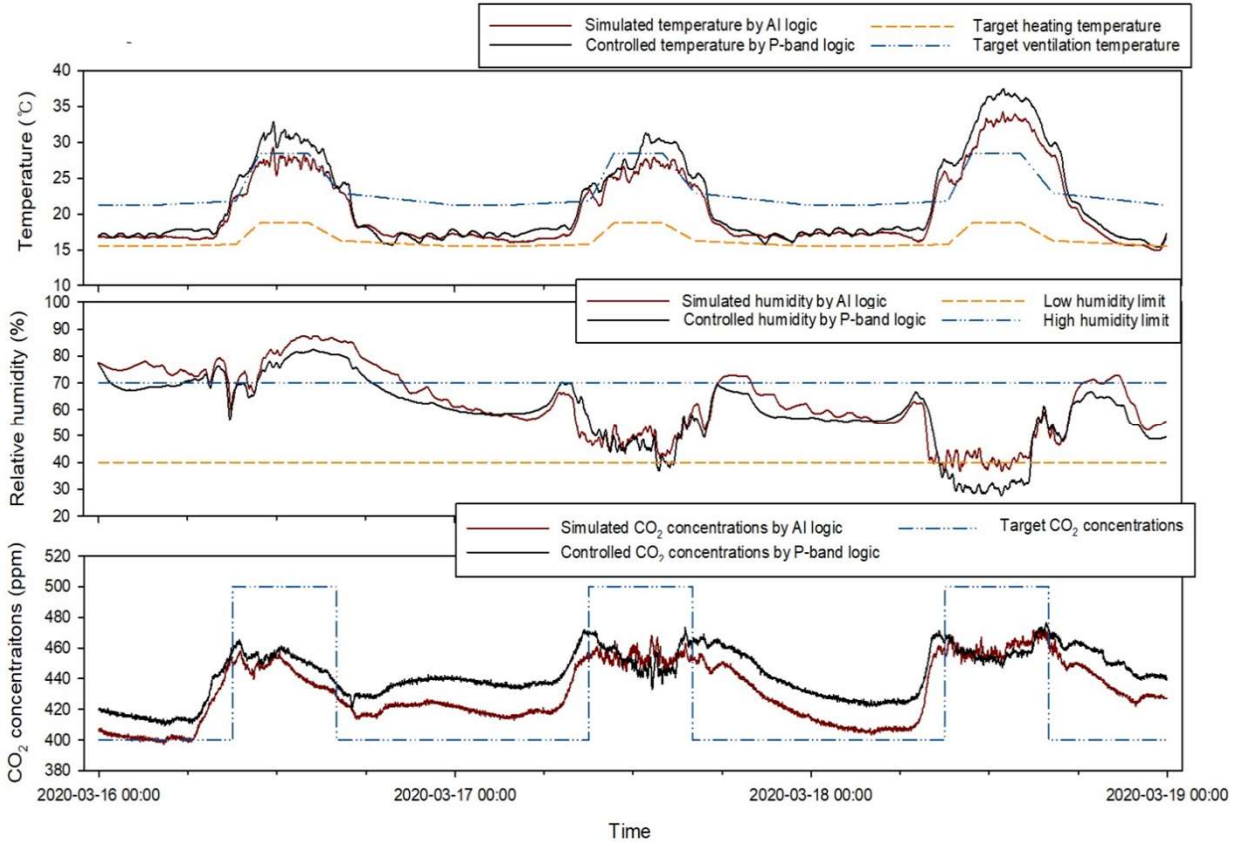

Figure 128. Comparison of the simulated control performance with AI and the actual controlled (P-band) performance for temperature, humidity, and CO_2 levels in spring.

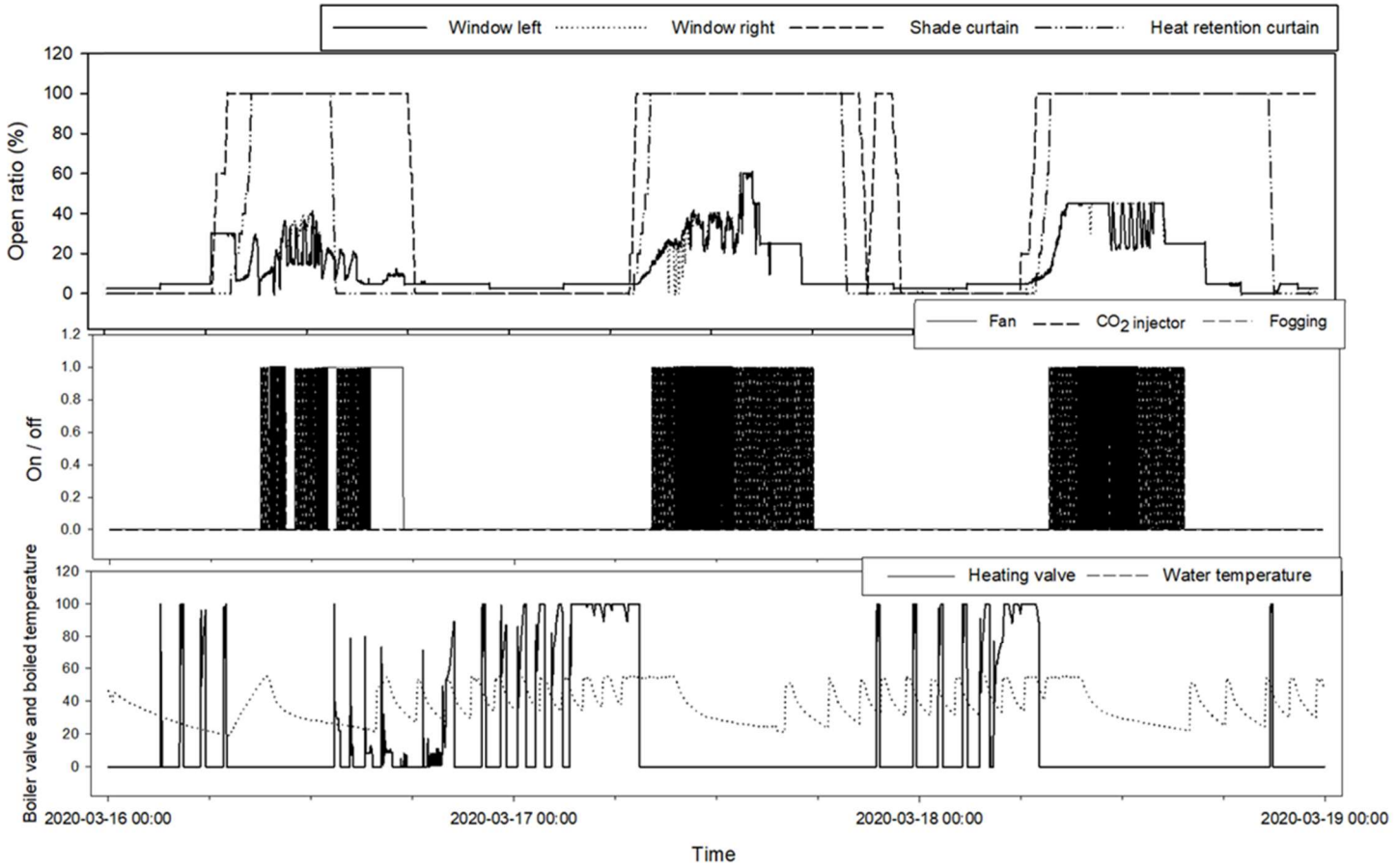

Figure 129. Control history for the actual P-band-controlled actuators inside the greenhouse in spring.

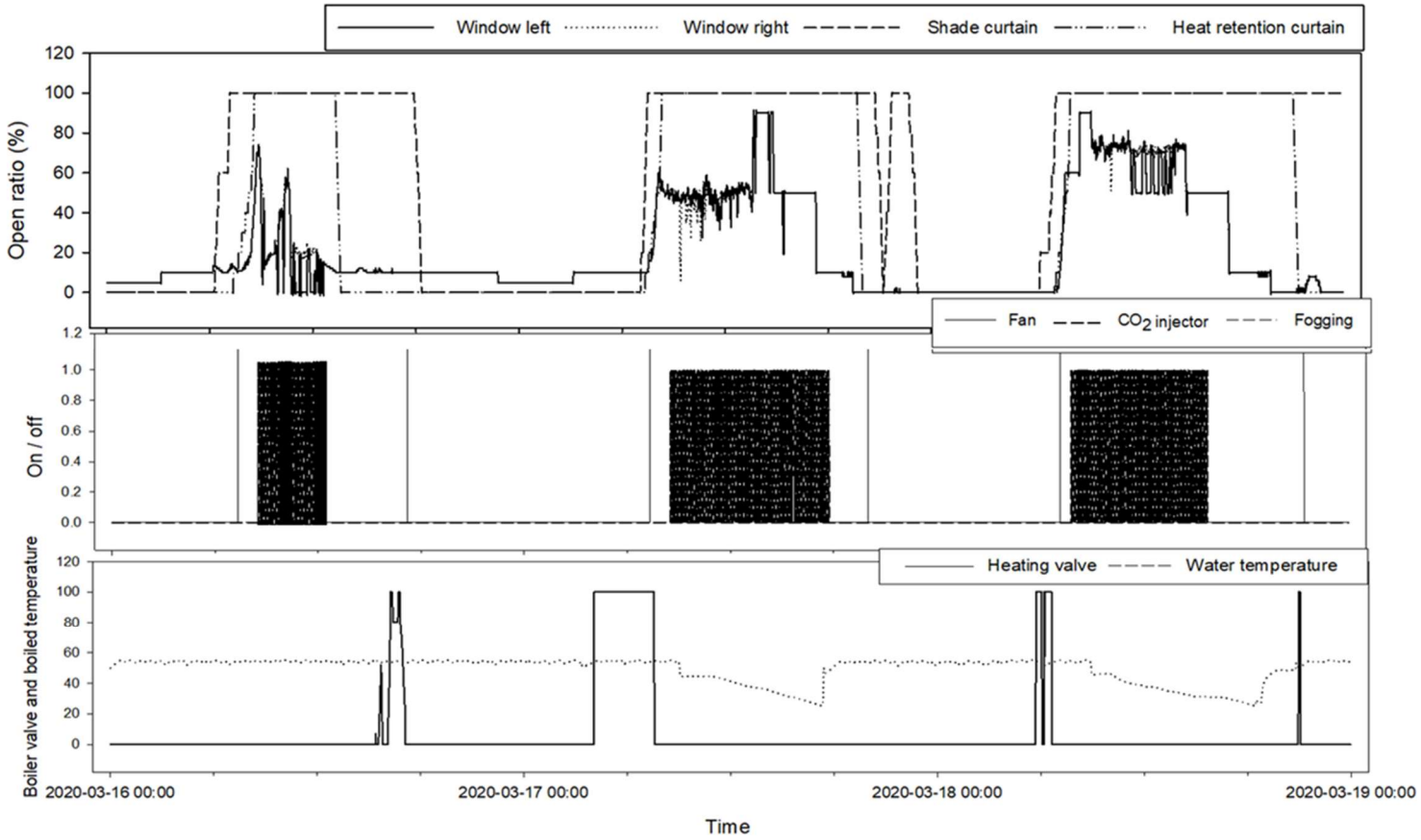

Figure 130. Control history for the simulated AI logic-controlled actuators inside the greenhouse in spring.

6.3.3. COMPARISON OF SIMULATION RESULTS IN SUMMER CLIMATE

Figure 133 presents the temperature, humidity, and CO_2 levels in summer for the actual PD-band control system and the simulated values for these for the proposed AI-based model. For temperature, the simulated AI-based control performance was better than the actual PD-band-controlled performance (RMSE of 3.21 and 3.45, respectively), while humidity was also better controlled by the AI logic (RMSE of 5.11 and 6.25, respectively). For CO_2, the RMSE for the simulated AI-based control was lower than that for the actual PD-band control system (40.04 and 41.92 ppm, respectively).

When comparing the energy consumption patterns, the use of fogging increased significantly, while heating was rarely used, appearing only in the simulated AI logic system. Ventilation using the windows continued to represent a major component of the energy expenditure. AI logic again limited the use of the fans (Figure 131). It was estimated that it cost 9,645 KRW to power the greenhouse under actual PD-band-based control, compared to 12,472 KRW for the AI logic simulation, an increase of 29.20% (Figure 132 and Figure 122). This was primarily due to the use of heating under the AI-based control system.

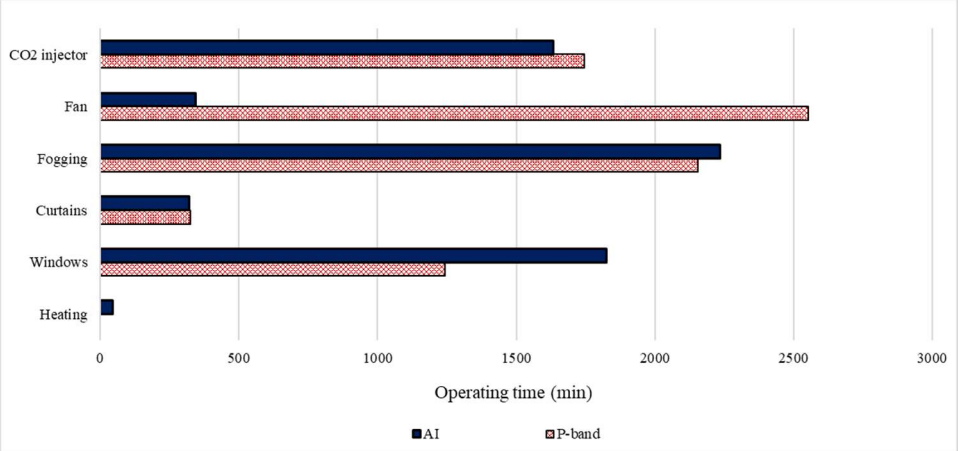

Figure 131. Operating times for the actuators over three days in summer.

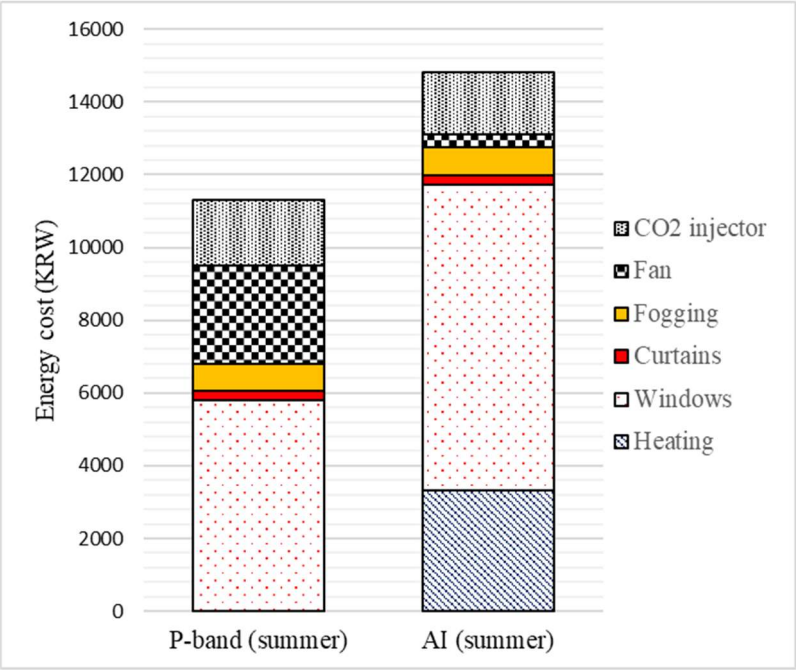

Figure 132. Operating costs for the actuators over three days in summer.

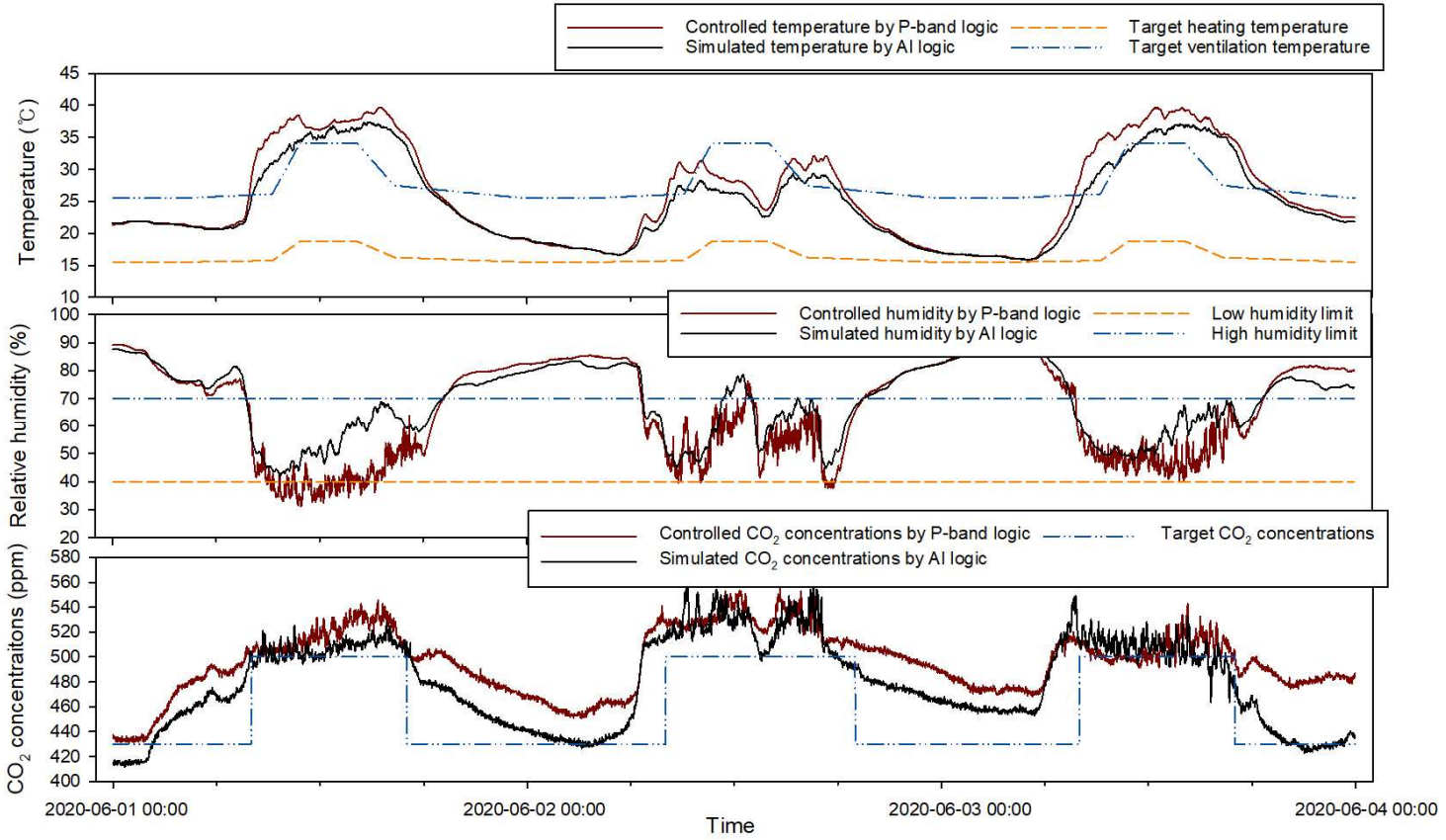

Figure 133. Comparison of the simulated control performance with AI logic and the actual P-band control performance for temperature, humidity, and CO_2 levels in summer.

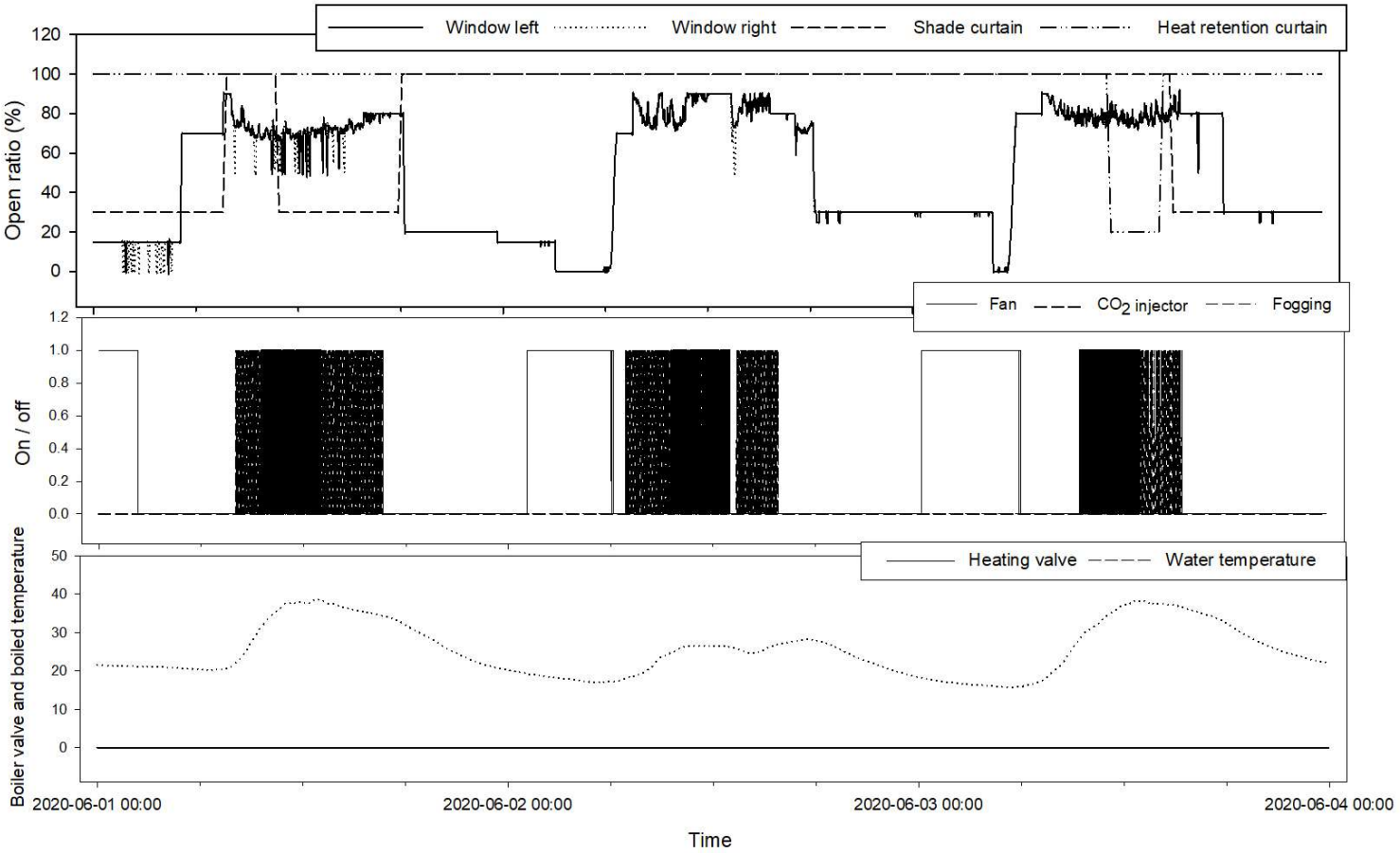

Figure 134. Control history for the actual P-band-controlled actuators inside the greenhouse in spring.

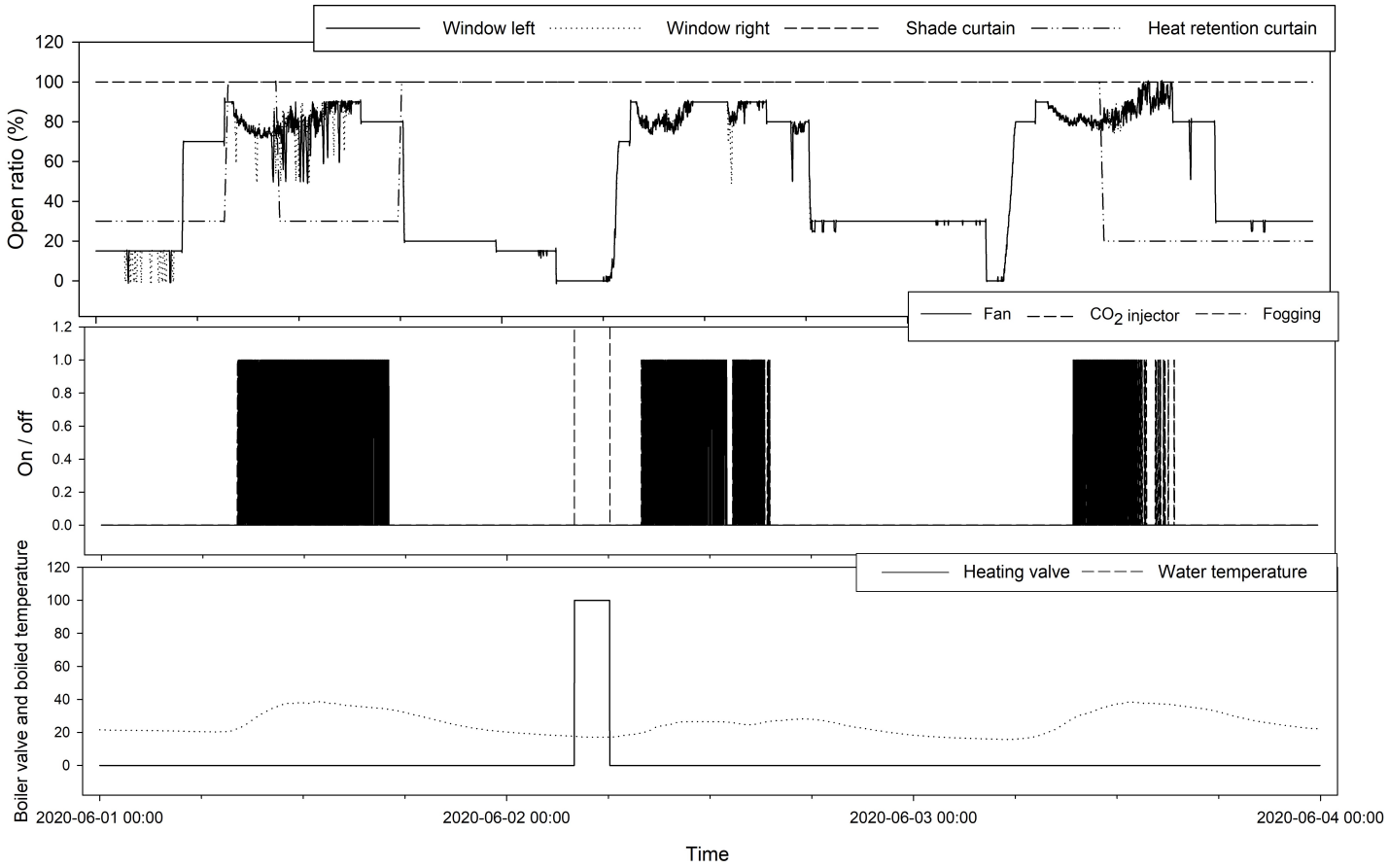

Figure 135. Control history for the simulated AI logic-controlled actuators inside the greenhouse in summer.

Table 36. The overall results of the simulation study.

	Temperature (℃)		Humidity (%)		CO_2 (ppm)		Energy (KRW)	
	P-band	AI	P-band	AI	P-band	AI	P-band	AI
Winter	1.85	2.01	4.23	3.22	57.22	34.55	81,182	72,579
Spring	2.23	1.94	4.22	3.45	35.01	32.11	56,367	45,216
Summer	3.45	3.21	6.25	5.11	41.92	40.04	9,645	12,472

6.4. FIELD TEST AND RESULTS

Field verification experiments were conducted using the algorithm proposed in this study over two weeks from March 16 to 31, 2020. Two tomato greenhouses in Gangneung were managed using a commercial controller and the proposed AI controller. As shown in Figure 136, the second greenhouse in a row of four Venlo greenhouses was equipped with the commercial controller, while the fourth greenhouse was equipped with the AI-based environmental control logic proposed in this study. The heating systems operated under the same conditions and were excluded from optimized empirical control because it was physically impossible to gain complete independent control in the test greenhouse.

The environment was run for two months using the linear algorithm before being run by the AI-based control logic. The CNN-LSTM model was in control for two weeks using the automatic model update system described above. Based on this, the final signal was determined using the OFNN Adadelta algorithm, and energy-saving logic was added by adjusting the temperature, humidity, and CO_2 concentration in the greenhouse.

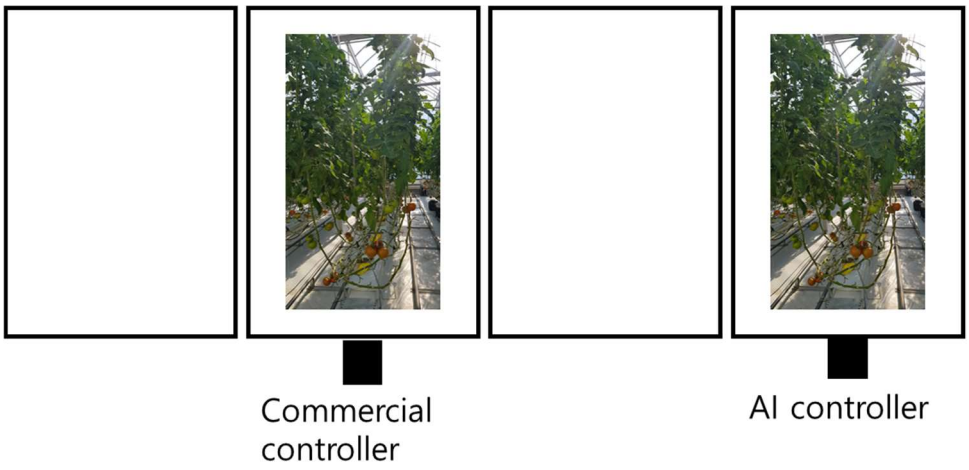

Figure 136. Tomato greenhouses used for the verification experiment.

The Adadelta algorithm is presented as Equation 43. The constants used were as follows: learning rate 0.003, rho 0.95, and epsilon 0.00001. Figure 137–Figure 139 show the changes in temperature, humidity, and CO_2 concentration during the experiment. The target temperature was set for each of the six periods. The minimum humidity was set at 50%, while 400 ppm and 470 ppm were set as the boundary concentrations for CO_2.

For the target temperature, the commercial controller exhibited an RMSE of 2.15 ℃, which was higher than the 1.78 ℃ for the AI controller (1.65 ℃ and 0.98 ℃ for the 30-min average; Figure 137). This means that the AI controller outperformed the commercial product in terms of temperature control. On the other hand, when comparing the control performance when the humidity control falls below 50%, RMSE 4.6% AI controller for commercial products is 6.5%, and half-hour change average is 5.2% for commercial products and 3.6% for AI controllers. Thus, for humidity control, the commercial controller exhibited better performance. On average, the fogging time was about 175 minutes for the commercial controller during the day, compared to 160 minutes for the AI controller, but there was no significant difference. This lower performance may be due to greater ventilation leading to the loss of humidity. In addition, the AI logic tends to inject less energy due to the energy consumption function, which was most apparent in the control of the CO_2 levels; for the target of 470 ppm, the commercial controller had an RMSE of 34.45 ppm, while the AI controller had an RMSE of 44.22 ppm (Figure 139).

These results indicate that adjusting the compensation value in the cost gate leads to favorable decisions with regards to the control of humidity and CO_2. In this experiment, under the assumption that the heating conditions were the same, it is applied to energy consumption only by driving the controller. It appears that the AI logic placed more weight on ventilation (Figure 140), while fan operation was minimized under AI control. In the model, the fans had no direct effect on environmental control. However, the fans circulated the air above and below the greenhouse to evenly distribute the elements of the environment. This was not considered by the AI logic, hence the implementation of additional restrictions.

In this test, the climate control logic for heating was excluded because, unlike other controllers, heating uses the same boiled water source unless a completely independent boiler is used. In other words, completely independent control is physically impossible, so it was excluded from consideration. This element of the greenhouse environment thus requires additional analysis through the use of a completely separate heating system in the future. Nevertheless, the predictive model proposed in this study was confirmed to be capable of achieving superior performance in terms of environmental control compared to the commercial controller in the field.

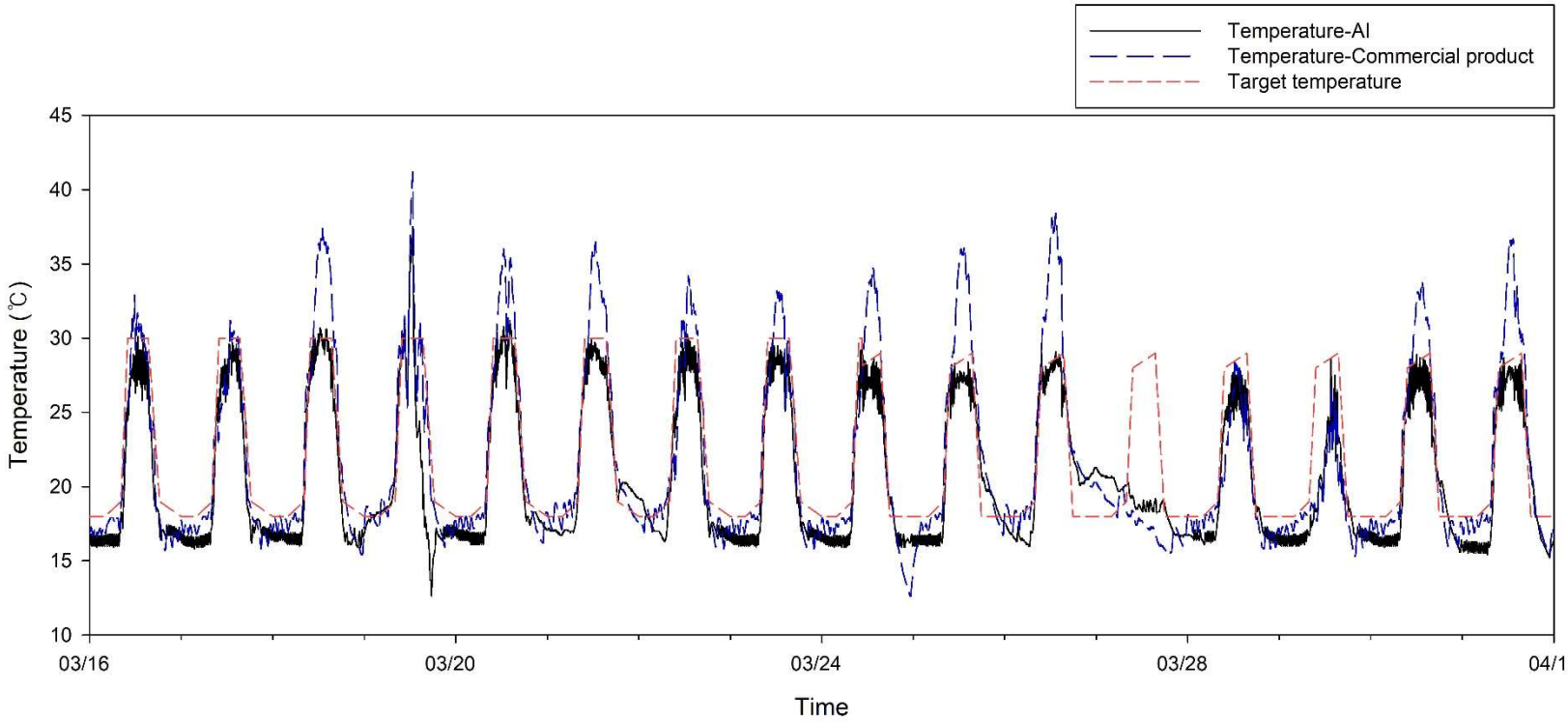

Figure 137. Test results for the AI logic and commercial controller in terms of temperature control.

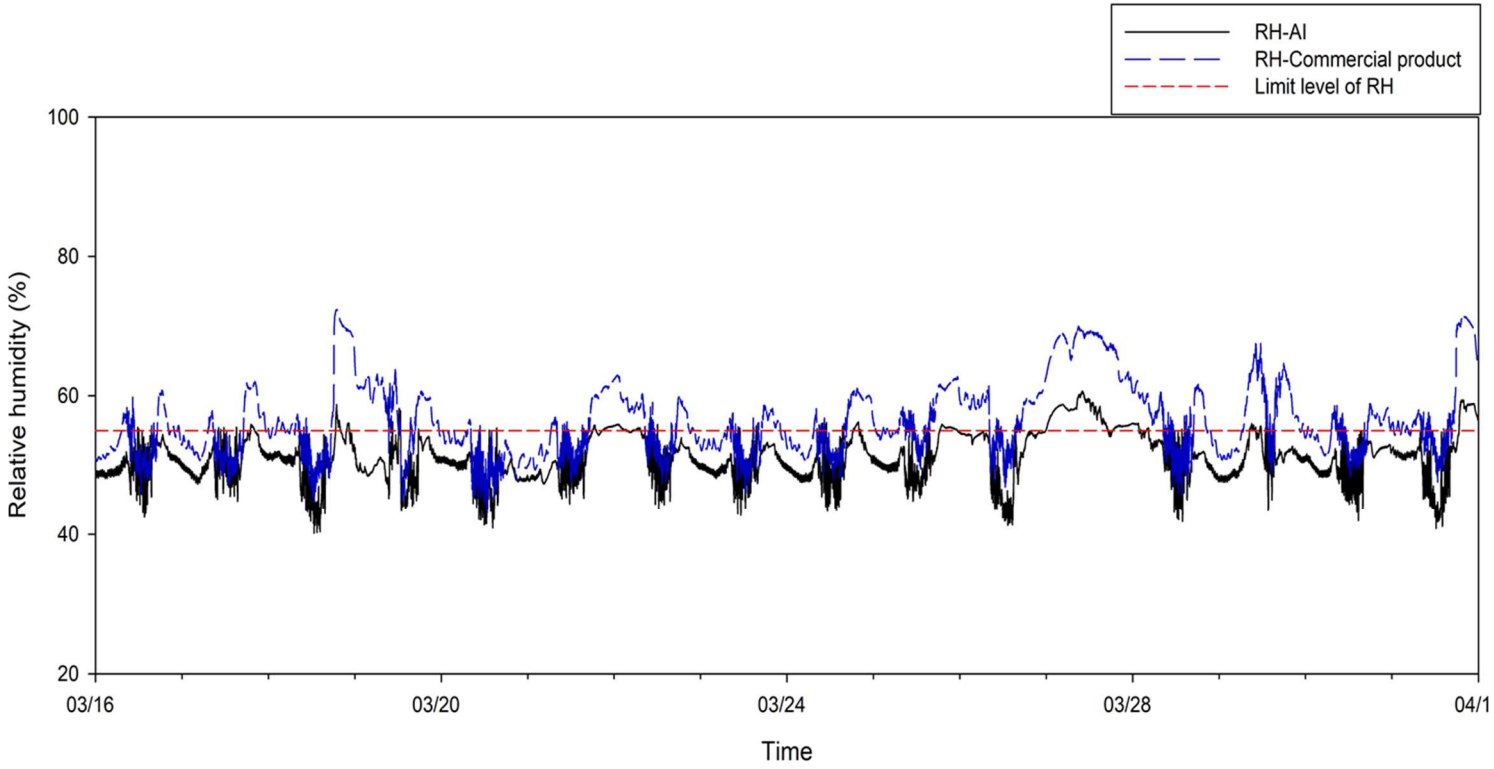

Figure 138. Test results for the AI logic and commercial controller in terms of humidity control.

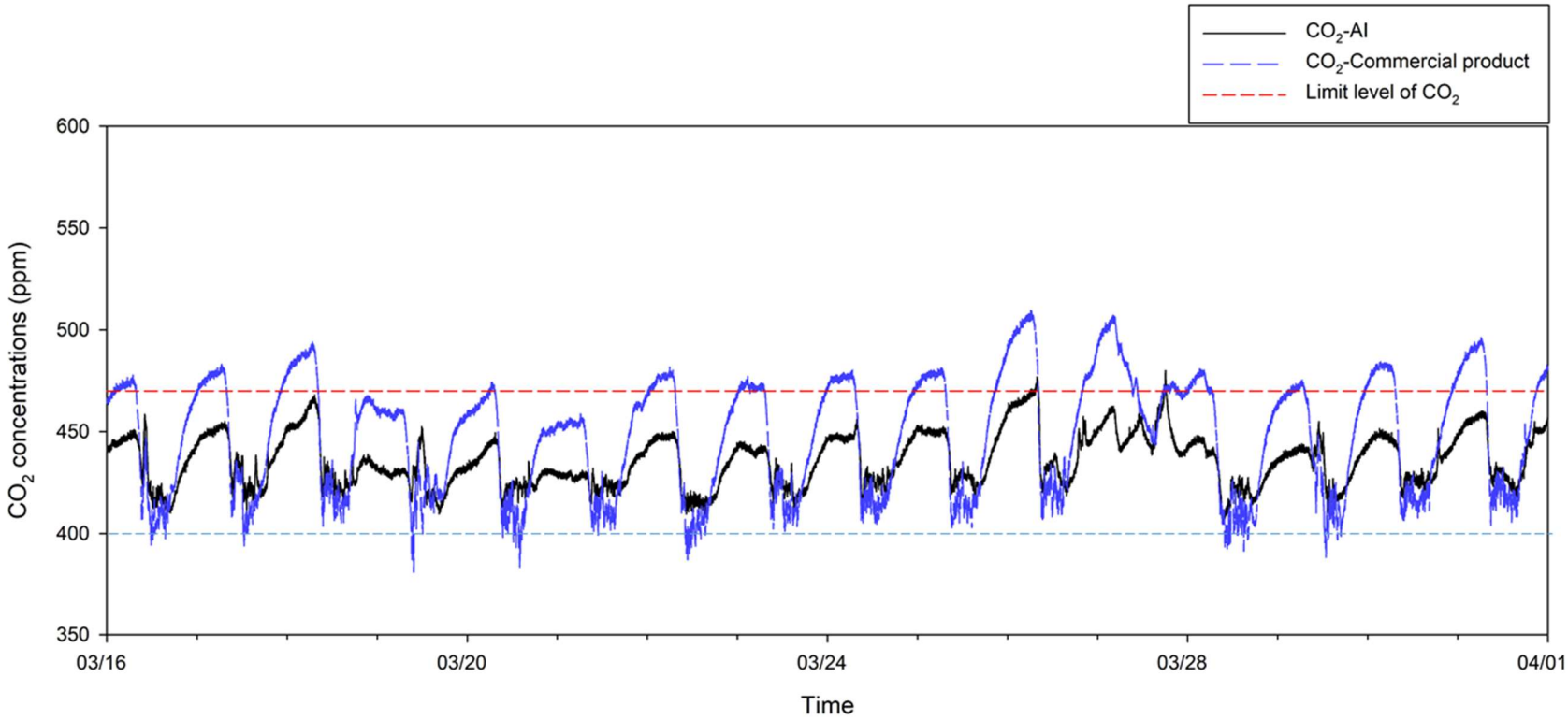

Figure 139. Test results for the AI logic and commercial controller in terms of CO_2 control.

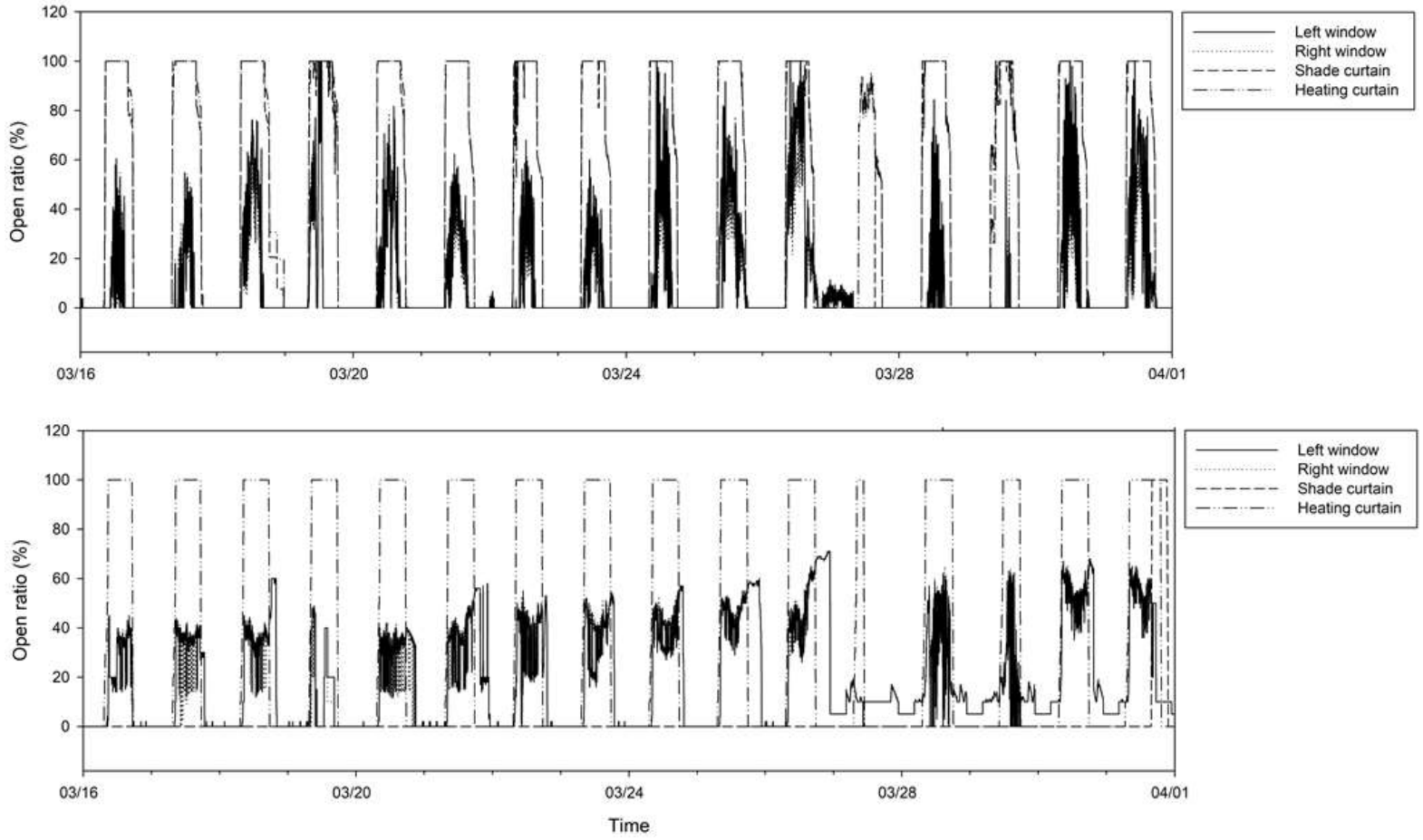

Figure 140. Window and curtain control history for the AI logic and commercial controller.

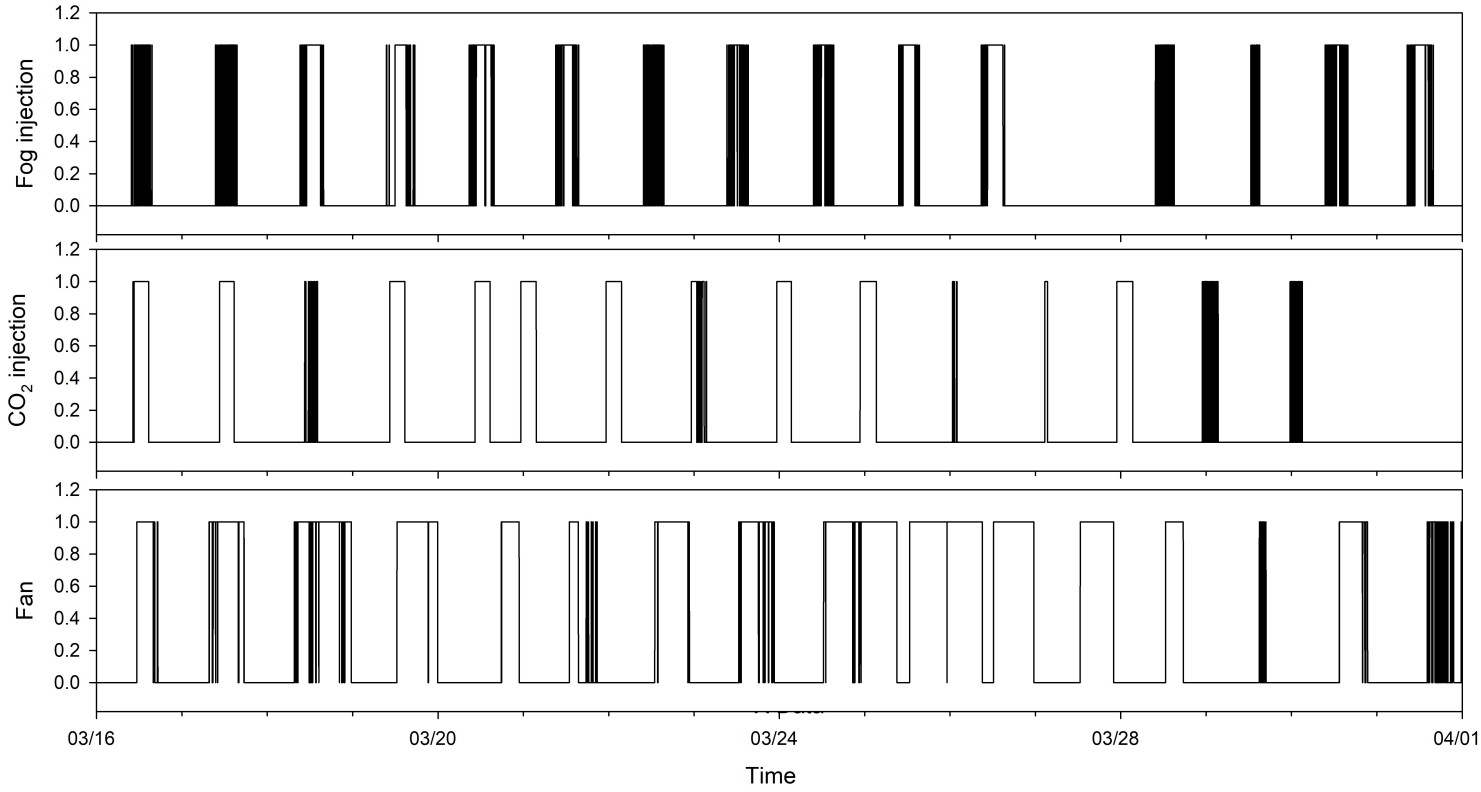

Figure 141. Fogging, CO_2 spraying, and fan control history for the commercial controller.

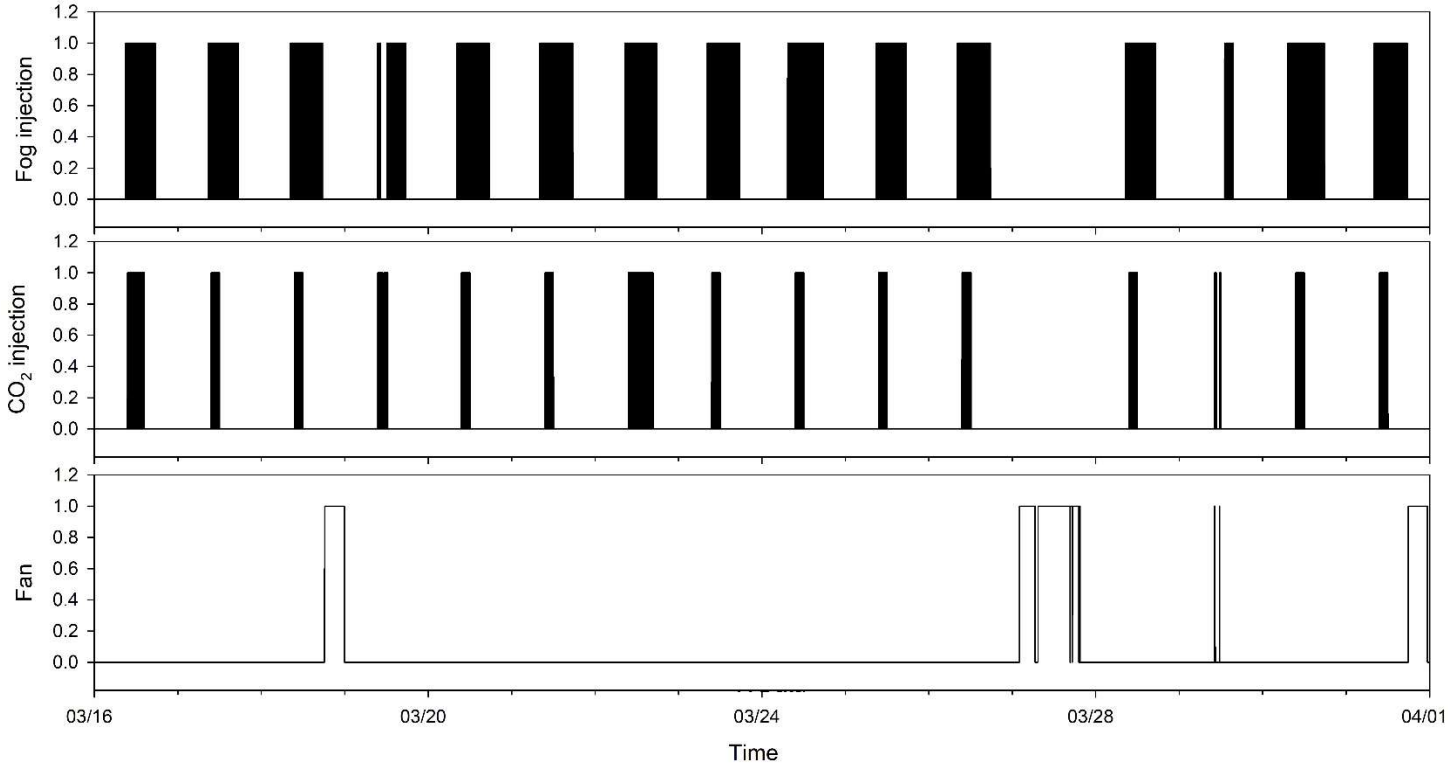

Figure 142. Fogging, CO_2 spraying, and fan control history for the AI controller.

6.5. CHAPTER CONCLUSION

This study proposed a control system based on an OFNN to determine the environmental control signals using a previously developed prediction model. This was achieved by determining the cost calculated from a comparison between the temperature obtained from the prediction model and target value and then developing a signal decision algorithm via optimization.

In order to reduce the energy consumption of the actuators, the operating costs for each actuator were functionalized and applied to the optimization algorithm, which led to more efficient operation as verified by three season-specific simulations. In winter, energy costs totaled 81,182 KRW in the conventionally controlled greenhouse, which was higher than the 72,579 KRW required for the AI logic-based system, a reduction of 10.59%. In spring, the actual costs were 56,367 KRW compared to 45,216 KRW for the simulated AI logic model (-19.78%). This cost reduction was primarily due to limitations on the use of the boiler and ventilation windows. However, the energy used in summer was higher for the AI-based model (12,472 KRW) than the actual PD-band-based control system (9,645 KRW), an increase of 29.20%.

The performance of the energy-saving environmental control logic based on the proposed automatic prediction model was verified using empirical field experiments over two weeks. Excellent results were obtained for temperature control when compared to a commercially available controller, and energy-savings of about 5.67% was achieved during this period. It is thus expected that the AI-based environmental control logic proposed in this study will improve the performance of multivariable greenhouse climate control.

7. OVERALL CONCLUSION AND DISCUSSION

This chapter reviews the results of this study, including its achievements and limitations, and discusses potential future research directions.

7.1. CONCLUSION

This study developed a high-performance controller that combined smart farm and internal environmental control within a greenhouse system, incorporating automation and ICT technology combined with AI to improve performance and increase the convenience for users. The methodology and technologies were specifically employed to provide an optimal control solution based on data from the Lohan greenhouse system. As such, greenhouse environmental control using AI big data was presented.

In commercial controllers, the general linear algorithm uses P-band-based environmental control technology. This P-band-based algorithm is designed to input empirical-based coefficients rather than mathematical or scientific system modeling to reflect the influence of simple external factors. In Chapter 3, the methodology for optimizing the coefficients in linear algorithm-based control was presented, and a controller-based design was demonstrated.

Chapter 4 presented data-based deep-learning modeling of environmental changes in a greenhouse. Many existing studies have developed several models based on scientific energy conservation laws; however, only a few have employed data-based deep learning. In this study, various training and verification sets were presented for each step to predict changes in the temperature, humidity, and CO_2 concentration of the greenhouse. The deep-learning models RNN-LSTM and CNN-LSTM, which have previously exhibited excellent predictive performance, were developed to predict the greenhouse environmental conditions. The models were successfully modified and their performance evaluated. The greenhouse environmental prediction model that employed a CNN was specifically designed for the prediction of physiological activity, such as evapotranspiration; thus, the results are expected to have an academic contribution.

An improvement in humidity prediction was exhibited by both proposed models,

which included the evapotranspiration rate and data from the root-zone and leaf sensors, with a SEP of 5.78% and 6.12% for the RNN-LSTM and CNN-LSTM models compared to 9.12% for a previously developed model. This means that the relative humidity in the greenhouse and the relative humidity in the greenhouse are closely related to each other. In addition, if more information related to moisture is collected from the greenhouse, humidity prediction can be improved even further. In the future, it seems to be an important case study in which the two methods complement each other as a hybrid concept between the deep learning model based on data and the physical model.

The deep-learning-based prediction model developed in his study was designed to respond to changes in the atmospheric environment according to operational changes. A method of determining the optimal actuator signal by backtracking was also introduced to the structure of the OFNN. This was employed in the calculation of the cost from the target environmental profile, current value, and environmental change that would occur after 30 min, and an optimization method to reduce this cost was devised. SGD, which has been utilized in previous machine-learning and deep-learning studies, was employed in the form of an OFNN. Moreover, a multiwindow ventilation control experiment found that the actuator signal was more sensitive to environmental changes than the existing linear algorithm.

To promote energy conservation, the driving energy for each actuator was used as a function in a cost gate that considered the energy consumption when determining the control signal. The actual energy-saving effect was confirmed using field testing, thus demonstrating great promise for use in various sectors in the near future.

The cost-based environmental control technology developed in this study has numerous potential applications. In particular, if data-based plant growth is modeled following the design presented in this study, the growth rate could be increased, or more informed decisions about the environmental components could be made, which would be favorable for the physiological activity of crops. It was confirmed that this framework could be employed in a system that analyzes the data obtained from various facilities and be used to make appropriate decisions in smart farm research.

7.2. DISCUSSION

P-band based control logic is commonly employed by commercialized greenhouse controllers (Kim et al., 2017) and requires users to input many settings to ensure optimal environmental conditions, which may not be easy for growers. In contrast, the predictive model and control logic developed in this study are trained using the greenhouse conditions without requiring settings to be input separately. For example, in the May 11 test (Figures 113 and 114), the actual temperature decreased sharply between 6 AM and 7 AM, caused by low-temperature air entering the greenhouse. The conventional controller based on P-band logic strongly considered ventilation due to the influence of increasing solar radiation after sunrise and opened several windows. This is because the simple linear algorithm made the decision to open the windows in response to a sharp rise in the temperature at this time. In contrast, the control method developed in this study was trained using the greenhouse data; thus, the drop in temperature due to windows opening was predictable, so the controller decided to keep the window closed at this time and opened it a little.

Climate control methods based on the actual applied environmental model have been regularly proposed, and it was a major trend of applying the simulation through system identification (Bennis et al., 2008; Shen et al., 2013). In this paper, the proposed method was implemented by the proposed optimization method, and the feasibility was confirmed using field testing. This method is expected to be applicable not only to greenhouse climate control but also to climate management for livestock facilities or environmental management of residential buildings. This study found that a neural-network-based prediction model and control logic yield better control signals only in the greenhouse in which the training was conducted. In addition, the logic was applied during spring and autumn, and window control was conducted only for ventilation. However, humidity and radiation are also important factors affected by ventilation. Therefore, various environmental factors and greenhouse structures should be investigated using the OFNN structure and cost function.

It was more difficult to obtain the same control over humidity as was achieved for temperature. Although it was possible to increase the predicted performance by

adding sensor information related to the amount of evapotranspiration or crop conditions, this requires additional sensors. In particular, the Stanghellini model was used as the evapotranspiration model, and the LAI was fixed at 2.5. The LAI for tomatoes is actually very difficult to measure non-destructively in a greenhouse. In this regard, it is necessary to apply a deep-learning algorithm that deals with image information such as the CNN-LSTM model, thus the proposed CNN-LSTM model and the two-dimensionalization of the climate information may provide the roadmap for future applications.

In addition, the CNN structure has several advantages over the general NN. Structurally, the advantage of CNN is that it can express the connectivity between input data well and can judge the importance of data in the early stages. Li and Liu, (2019) employed IB theory to understand the dynamic behavior of CNNs and investigate how the fundamental features have impact on the performance of CNNs. Through a series of experimental analysis on benchmark of MNIST, the study demonstrated that the compression phase is not observed in all these cases. This shows us the CNNs have a rather complicated behavior than feedforward neural networks. In addition, the climate prediction model to be applied to greenhouses needs to be studied in a structure that includes image data of crops in the future. As a starting point for these studies, the deep neural network model based on CNN structure will be one of possible solution.

In addition to optimizing a predictive model using the OFNN proposed in this study to determine the control signal, reinforcement learning (RL) has also been used as a climate control technique based on AI (Vinyals et al., 2019). RL is a model-free framework for solving optimal control problems known as Markov decision processes (MDPs) (Puterman, 2014). Buşoniu et al. (2018) reported that MDPs work in discrete time: at each time step, the controller receives feedback from the system in the form of a state signal and takes an action in response. Hence, the decision rule is a state feedback control law, called a policy in RL. The action changes the system state, possibly in a stochastic manner, and the latest transition is evaluated via a reward function (i.e., negative cost). The optimal control objective is then to maximize from each initial state the (expected) cumulative reward, known as the value. In this respect, it differs from the OFNN proposed in this study. The OFNN

looks to reduce this by calculating a penalty (cost) in the action decision, and in RL, the action is decided by looking to obtain greater benefit by considering the predicted and cumulative rewards. If RL is used to control the climate of a greenhouse, environmental behavior can be designed to compensate for crop physiology or growth. Nevertheless, there is a chance that it will be negative, and its prospects for use in the field are not optimistic. This method is likely to lead to unstable behavior before the model is fully established due to the nature of the RL algorithm. Therefore, the environment in which the RL algorithm is most effective is virtual reality, such as in a video game. Further discussion on the potential for RL-related greenhouse control and its design is described in the appendix of this thesis.

7.2.1. LIMITATIONS AND FUTURE WORK

Although not covered in this study, a significant component of environmental control inside a greenhouse is the setting of the target environmental values. It is crucial that the target environment can be flexibly changed, taking into account the agronomic view and the crop conditions. In this model design, how to determine the optimal conditions for the operation of the controller by reflecting the environmental factors in the input is important. For example, in the field, an approach to increasing the growth rate or increasing the daily difference by determining the thickness of the plant stem is required (Qian et al., 2015), and it would be desirable to utilize image information and match it to the climate control history to focus on crop growth (Dong et al., 2017; Gruber et al., 2011). It is possible that the implementation of this study could be conducted using a framework based on RL, but there have been very few cases in which it has been used as a concrete design for a climate control system that considers plant growth.

7.2.2. THE POSSIBILITY OF UNKNOWN SIDE EFFECTS USING AI LOGIC

A cautious approach to the use of AI algorithms has been suggested by some researchers in previous studies (Amodei et al., 2016; J. Chen et al., 2015; Wilks,

2019). In particular, careful consideration of potential unusual behavior in the pursuit of a particular purpose is required. In this study, an AI algorithm with two approaches were proposed for greenhouse environmental control. The algorithm considered the target values for environmental control and the energy consumption rate of the actuators, while the optimization of the driver in issuing a command was restricted. Therefore, potentially dangerous control was avoided, and this was confirmed through simulations. Nevertheless, there are cases in which signal decisions are made under fitted characteristics of the seasonal actuators, so it seems to be more stable when restrictive rules are added.

In addition, if the performance of a prediction model does not reach a certain level because the underlying algorithm is limited, or if the driving characteristics of the actuator use a unilateral learning set, it is unlikely that a correct decision will be made during the optimization process. In addition, if the constant ratios of the two targets (energy consumption and climate control) do not match, there is a concern that extreme control will occur. If there are no specific rules in a nonlinear model, sufficient simulation studies and feasibility testing should be conducted before using an AI algorithm in the field.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D., 2016. Concrete problems in AI safety. arXiv Prepr. arXiv1606.06565.

Anapalli, S.S., Ahuja, L.R., Gowda, P.H., Ma, L., Marek, G., Evett, S.R., Howell, T.A., 2016. Simulation of crop evapotranspiration and crop coefficients with data in weighing lysimeters. Agric. Water Manag. 177, 274–283. https://doi.org/https://doi.org/10.1016/j.agwat.2016.08.009

Ban, B., Kim, S., 2017. Control of nonlinear, complex and black-boxed greenhouse system with reinforcement learning, in: Information and Communication Technology Convergence (ICTC), 2017 International Conference On. IEEE, pp. 913–918.

Bartzanas, T., Boulard, T., Kittas, C., 2002. Numerical simulation of the airflow and temperature distribution in a tunnel greenhouse equipped with insect-proof screen in the openings. Comput. Electron. Agric. 34, 207–221. https://doi.org/http://dx.doi.org/10.1016/S0168-1699(01)00188-0

Baumeister, T., Brunton, S.L., Kutz, J.N., 2017. Deep Learning and Model Predictive Control for Self-Tuning Mode-Locked Lasers.

Benni, S., Tassinari, P., Bonora, F., Barbaresi, A., Torreggiani, D., 2016a. Efficacy of greenhouse natural ventilation: Environmental monitoring and CFD simulations of a study case. Energy Build. 125, 276–286. https://doi.org/10.1016/j.enbuild.2016.05.014

Benni, S., Tassinari, P., Bonora, F., Barbaresi, A., Torreggiani, D., 2016b. Efficacy of greenhouse natural ventilation: Environmental monitoring and CFD simulations of a study case. Energy Build. 125, 276–286. https://doi.org/http://dx.doi.org/10.1016/j.enbuild.2016.05.014

Bennis, N., Duplaix, J., En??a, G., Haloua, M., Youlal, H., 2008. Greenhouse climate modelling and robust control. Comput. Electron. Agric. 61, 96–107. https://doi.org/10.1016/j.compag.2007.09.014

Betiku, E., Taiwo, A.E., 2015. Modeling and optimization of bioethanol production from breadfruit starch hydrolyzate vis-à-vis response surface methodology and artificial neural network. Renew. Energy 74, 87–94. https://doi.org/https://doi.org/10.1016/j.renene.2014.07.054

Beven, K., 1979. A sensitivity analysis of the Penman-Monteith actual evapotranspiration estimates. J. Hydrol. 44, 169–190.

Blasco, X., Martínez, M., Herrero, J.M., Ramos, C., Sanchis, J., 2007. Model-based predictive control of greenhouse climate for reducing energy and water consumption. Comput. Electron. Agric. 55, 49–70. https://doi.org/10.1016/j.compag.2006.12.001

Buşoniu, L., de Bruin, T., Tolić, D., Kober, J., Palunko, I., 2018. Reinforcement learning for control: Performance, stability, and deep approximators. Annu. Rev. Control 46, 8–28. https://doi.org/https://doi.org/10.1016/j.arcontrol.2018.09.005

Calise, A.J., Hovakimyan, N., Idan, M., 2001. Adaptive output feedback control of nonlinear systems using neural networks. Automatica 37, 1201–1211.

Cha, B., Choi, M., Kim, B., Cheon, O., Han, T., Kim, J., Park, S., 2016. Research of Next Generation IoF-Cloud based Smart Geenhouse & Services. Smart Media J. 5, 17–24.

Chen, B., Zhang, H., Lin, C., 2015. Observer-based adaptive neural network control for nonlinear systems in nonstrict-feedback form. IEEE Trans. neural networks Learn. Syst. 27, 89–98.

Chen, J., Huang, T.-C., 2004. Applying neural networks to on-line updated PID controllers for nonlinear process control. J. Process Control 14, 211–230.

Chen, J., Xu, F., Tan, D., Shen, Z., Zhang, L., Ai, Q., 2015. A control method for agricultural greenhouses heating based on computational fluid dynamics and energy prediction model. Appl. Energy 141, 106–118. https://doi.org/http://dx.doi.org/10.1016/j.apenergy.2014.12.026

Chiew, F.H.S., Kamaladasa, N.N., Malano, H.M., McMahon, T.A., 1995. Penman-Monteith, FAO-24 reference crop evapotranspiration and class-A pan data in Australia. Agric. Water Manag. 28, 9–21.

Coelho, J.P., de Moura Oliveira, P.B., Cunha, J.B., 2005. Greenhouse air temperature predictive control using the particle swarm optimisation algorithm. Comput. Electron. Agric. 49, 330–344. https://doi.org/https://doi.org/10.1016/j.compag.2005.08.003

Dariouchy, A., Aassif, E., Lekouch, K., Bouirden, L., Maze, G., 2009. Prediction of the intern parameters tomato greenhouse in a semi-arid area using a time-series model of artificial neural networks. Measurement 42, 456–463. https://doi.org/http://dx.doi.org/10.1016/j.measurement.2008.08.013

Diaconescu, E., 2008. The use of NARX neural networks to predict chaotic time series. Wseas Trans. Comput. Res. 3, 182–191.

Ding, D., Wang, Z., Han, Q., Wei, G., 2019. Neural-Network-Based Output-Feedback Control Under Round-Robin Scheduling Protocols. IEEE Trans. Cybern. 49, 2372–2384. https://doi.org/10.1109/TCYB.2018.2827037

Ding, Y., Wang, L., Li, Y., Li, D., 2018. Model predictive control and its application in agriculture: A review. Comput. Electron. Agric. 151, 104–117. https://doi.org/https://doi.org/10.1016/j.compag.2018.06.004

Dodge, S., 2019. A Data Science Framework for Movement. Geogr. Anal.

Dong, T., Liu, J., Qian, B., Jing, Q., Croft, H., Chen, J., Wang, J., Huffman, T., Shang, J., Chen, P., 2017. Deriving Maximum Light Use Efficiency from Crop Growth Model and Satellite Data to Improve Crop Biomass Estimation. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10, 104–117. https://doi.org/10.1109/JSTARS.2016.2605303

El Ghoumari, M.Y., Tantau, H.J., Serrano, J., 2005. Non-linear constrained MPC: Real-time implementation of greenhouse air temperature control. Comput. Electron. Agric. 49, 345–356. https://doi.org/10.1016/j.compag.2005.08.005

Fan, R., Yang, X., Xie, H., Reeb, M.-A., 2012. Determination of nutrients in hydroponic solutions using mid-infrared spectroscopy. Sci. Hortic. (Amsterdam). 144, 48–54.

Fan, Y., Lu, X., Li, D., Liu, Y., 2016. Video-based emotion recognition using CNN-RNN and C3D hybrid networks, in: Proceedings of the 18th ACM International Conference on Multimodal Interaction. pp. 445–450.

Ferreira, P.M., Faria, E.A., Ruano, A.E., 2002. Neural network models in greenhouse air temperature prediction. Neurocomputing 43, 51–75.

Fitz-Rodríguez, E., Kacira, M., Villarreal-Guerrero, F., Giacomelli, G.A., Linker, R., Kubota, C., Arbel, A., 2012. Neural network predictive control in a naturally ventilated and fog cooled greenhouse. Acta Hortic.

Fitz-Rodríguez, E., Kubota, C., Giacomelli, G.A., Tignor, M.E., Wilson, S.B., McMahon, M., 2010. Dynamic modeling and simulation of greenhouse environments under several scenarios: A web-based application. Comput. Electron. Agric. 70, 105–116. https://doi.org/https://doi.org/10.1016/j.compag.2009.09.010

Fourati, F., Chtourou, M., 2007a. A greenhouse control with feed-forward and recurrent neural networks. Simul. Model. Pract. Theory 15, 1016–1028. https://doi.org/http://dx.doi.org/10.1016/j.simpat.2007.06.001

Fourati, F., Chtourou, M., 2007b. A greenhouse control with feed-forward and recurrent neural networks. Simul. Model. Pract. Theory 15, 1016–1028. https://doi.org/10.1016/j.simpat.2007.06.001

Francik, S., Kurpaska, S., 2020. The Use of Artificial Neural Networks for Forecasting of Air Temperature inside a Heated Foil Tunnel. Sensors 20, 652.

Frausto, H.U., Pieters, J.G., Deltour, J.M., 2003. Modelling Greenhouse Temperature by means of Auto Regressive Models. Biosyst. Eng. 84, 147–157. https://doi.org/http://dx.doi.org/10.1016/S1537-5110(02)00239-8

Gonzaga, J.C.B., Meleiro, L.A.C., Kiang, C., Maciel Filho, R., 2009. ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. Comput. Chem. Eng. 33, 43–49. https://doi.org/https://doi.org/10.1016/j.compchemeng.2008.05.019

Gouadria, F., Sbita, L., Sigrimis, N., 2017. A greenhouse system control based on a PSO tuned PI regulator, in: International Conference on Green Energy and Conversion Systems, GECS 2017. https://doi.org/10.1109/GECS.2017.8066235

Grinblat, G.L., Uzal, L.C., Larese, M.G., Granitto, P.M., 2016. Deep learning for plant identification using vein morphological patterns. Comput. Electron. Agric. 127, 418–424. https://doi.org/10.1016/j.compag.2016.07.003

Gruber, J.K., Guzmán, J.L., Rodríguez, F., Bordons, C., Berenguel, M., Sánchez, J.A., 2011. Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation. Control Eng. Pract. 19, 354–366. https://doi.org/https://doi.org/10.1016/j.conengprac.2010.12.004

Han, S., Li, Y., Zhou, B., Liu, Z., Feng, J., Xiao, Y., 2019. An in-situ accelerated experimental testing method for drip irrigation emitter clogging with inferior water. Agric. Water Manag. 212, 136–154. https://doi.org/10.1016/j.agwat.2018.08.024

He, F., Ma, C., 2010. Modeling greenhouse air humidity by means of artificial neural network and principal component analysis. Comput. Electron. Agric. 71. https://doi.org/10.1016/j.compag.2009.07.011

Hemming, S., de Zwart, F., Elings, A., Righini, I., Petropoulou, A., 2019. Remote Control of Greenhouse Vegetable Production with Artificial Intelligence—Greenhouse Climate, Irrigation, and Crop Production. Sensors 19, 1807.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Hong, S., Lee, I., 2014. Predictive Model of Micro-Environment in a Naturally Ventilated Greenhouse for a Model-Based Control Approach. Prot. Hortic. Plant Fact. 23, 181–191.

Hongkang, W., Li, L., Yong, W., Fanjia, M., Haihua, W., Sigrimis, N.A., 2018. Recurrent Neural Network Model for Prediction of Microclimate in Solar Greenhouse. IFAC-PapersOnLine 51, 790–795. https://doi.org/https://doi.org/10.1016/j.ifacol.2018.08.099

Jeong, J.-H., Lim, C.-M., Jo, J.-H., Kim, J., Kim, S.-H., Lee, K.-Y., Lee, S.-S., 2019. A Study on the Monitoring System of Growing Environment Department for Smart Farm. J. Korea Inst. Information, Electron. Commun. Technol. 12, 290–298.

Jung, D.-H., Kim, H.-J., Kim, S.H., Choi, J., Kim, D.J., Park, H.S., 2019. Fusion of Spectroscopy and Cobalt Electrochemistry Data for Estimating Phosphate Concentration in Hydroponic Solution. Sensors . https://doi.org/10.3390/s19112596

Kamilaris, Andreas, Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90.

Kamilaris, A, Prenafeta-Boldú, F.X., 2018. A review of the use of convolutional neural networks in agriculture. J. Agric. Sci. 156, 312–322.

Kamp, P.G.H., 1996. Computerized environmental control in greenhouses: a step by step approach.

Khashei, M., Bijari, M., 2010. An artificial neural network (p,d,q) model for timeseries forecasting. Expert Syst. Appl. 37, 479–489. https://doi.org/http://dx.doi.org/10.1016/j.eswa.2009.05.044

Kim, B.S., Kang, B.G., Choi, S.H., Kim, T.G., 2017. Data modeling versus simulation modeling in the big data era: case study of a greenhouse control system. Simulation 93, 579–594. https://doi.org/10.1177/0037549717692866

Kim, D., Oh, H.-S., Moon, I.-C., 2019. Black-box Modeling for Aircraft Maneuver Control with Bayesian Optimization. Int. J. Control. Autom. Syst. 17, 1558–1568.

Kim, S., Lee, S., Park, K., Ryu, K., 2018. Prediction model of internal temperature using backpropagation algorithm for climate control in greenhouse. Hortic. Sci. Technol. 36, 713–729.

Kim, T.-Y., Cho, S.-B., 2019. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 182, 72–81. https://doi.org/https://doi.org/10.1016/j.energy.2019.05.230

King, A., 2017. The future of agriculture. Nature 544, S21–S23.

Kishor, N., Singh, S.P., 2007. Simulated response of NN based identification and predictive control of hydro plant. Expert Syst. Appl. 32, 233–244. https://doi.org/http://dx.doi.org/10.1016/j.eswa.2005.11.026

Lanfang, P., Wanliang, W., Qidi, W., 2000. Application of adaptive fuzzy logic system to model for greenhouse climate, in: Intelligent Control and Automation, 2000. Proceedings of the 3rd World Congress On. IEEE, pp. 1687–1691.

Lapedes, A., Farber, R., 1987. Nonlinear signal processing using neural networks: Prediction and system modelling.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., 2006. A tutorial on energy-based learning. Predict. Struct. data 1.

Lee, C.K., Chung, M., Shin, K.-Y., Im, Y.-H., Yoon, S.-W., 2019. A Study of the Effects of Enhanced Uniformity Control of Greenhouse Environment Variables on Crop Growth. Energies 12, 1749.

Lee, K.-H., Choi, Y.-S., Baek, S.-W., Jeong, J.-T., 2016. Smart Farm Technology for Production of Ornamental Plants in Future TT - Smart Farm Technology for Production of Ornamental Plants in Future, in: HORTICULTURE ABSTRACTS. Korean Society For Horticultural Science, pp. 35–36.

Li, J., Liu, D., 2019. Information Bottleneck Methods on Convolutional Neural Networks. arXiv Prepr. arXiv1911.03722.

Li, T.-S., Wang, D., Feng, G., Tong, S.-C., 2009. A DSC approach to robust adaptive NN tracking control for strict-feedback nonlinear systems. IEEE Trans. Syst. man, Cybern. part b 40, 915–927.

Li, X., He, Y., Fang, H., 2007. Non-destructive discrimination of Chinese bayberry varieties using Vis/NIR spectroscopy. J. Food Eng. 81, 357–363. https://doi.org/10.1016/J.JFOODENG.2006.10.033

Libardi, L.G.P., de Faria, R.T., Dalri, A.B., de Souza Rolim, G., Palaretti, L.F., Coelho, A.P., Martins, I.P., 2019. Evapotranspiration and crop coefficient (Kc) of pre-sprouted sugarcane plantlets for greenhouse irrigation management. Agric. Water Manag. 212, 306–316. https://doi.org/10.1016/j.agwat.2018.09.003

Livieris, I.E., Pintelas, E., Pintelas, P., 2020. A CNN–LSTM model for gold price time-series forecasting. Neural Comput. Appl. 1–10.

Lu, W., Pedrycz, W., Liu, X., Yang, J., Li, P., 2014. The modeling of time series based on fuzzy information granules. Expert Syst. Appl. 41, 3799–3808. https://doi.org/http://dx.doi.org/10.1016/j.eswa.2013.12.005

Maher, A., Kamel, E., Enrico, F., Atif, I., Abdelkader, M., 2016. An intelligent system for the climate control and energy savings in agricultural greenhouses. Energy Effic. 9, 1241–1255. https://doi.org/10.1007/s12053-015-9421-8

Menezes Jr, J.M.P., Barreto, G.D.A., 2006. A New Look at Nonlinear Time Series Prediction with NARX Recurrent Neural Network. SBRN 6, 28–33.

Mitrea, C.A., Lee, C.K.M., Wu, Z., 2009. A Comparison between Neural Networks and Traditional Forecasting Methods: A Case Study. Int. J. Eng. Bus. Manag. 1, 11. https://doi.org/10.5772/6777

Mohsenipour, M., Ebadollahi, M., Rostamzadeh, H., Amidpour, M., 2020. Design and evaluation of a solar-based trigeneration system for a nearly zero energy greenhouse in arid region. J. Clean. Prod. 254, 119990.

Montoya, A.P., Guzmán, J.L., Rodríguez, F., Sánchez-Molina, J.A., 2016. A hybrid-controlled approach for maintaining nocturnal greenhouse temperature: Simulation study. Comput. Electron. Agric. 123, 116–124. https://doi.org/10.1016/j.compag.2016.02.014

Moon, T., Ahn, T.I., Son, J.E., 2018. Forecasting Root-Zone Electrical Conductivity of Nutrient Solutions in Closed-Loop Soilless Cultures via a Recurrent Neural Network Using Environmental and Cultivation Information. Front. Plant Sci. 9, 859. https://doi.org/10.3389/fpls.2018.00859

Morota, G., Ventura, R. V, Silva, F.F., Koyama, M., Fernando, S.C., 2018. BIG DATA ANALYTICS AND PRECISION ANIMAL AGRICULTURE SYMPOSIUM: Machine learning and data mining advance predictive big data analysis in precision animal agriculture1. J. Anim. Sci. 96, 1540–1550. https://doi.org/10.1093/jas/sky014

Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., Nillaor, P., 2019. IoT and agriculture data analysis for smart farm. Comput. Electron. Agric. 156, 467–474. https://doi.org/https://doi.org/10.1016/j.compag.2018.12.011

Norton, T., Sun, D.-W., Grant, J., Fallon, R., Dodd, V., 2007. Applications of computational fluid dynamics (CFD) in the modelling and design of ventilation systems in the agricultural industry: A review. Bioresour. Technol. 98, 2386–2414. https://doi.org/10.1016/j.biortech.2006.11.025

Nury, A.H., Hasan, K., Alam, M.J. Bin, 2017. Comparative study of wavelet-ARIMA and wavelet-ANN models for temperature time series data in northeastern Bangladesh. J. King Saud Univ. - Sci. 29, 47–61. https://doi.org/http://dx.doi.org/10.1016/j.jksus.2015.12.002

Orgaz, F., Fernández, M.D., Bonachela, S., Gallardo, M., Fereres, E., 2005. Evapotranspiration of horticultural crops in an unheated plastic greenhouse. Agric. Water Manag. 72, 81–96. https://doi.org/https://doi.org/10.1016/j.agwat.2004.09.010

Pandey, S.K., Janghel, R.R., 2019. Recent Deep Learning Techniques, Challenges and Its Applications for Medical Healthcare System: A Review. Neural Process. Lett. https://doi.org/10.1007/s11063-018-09976-2

Park, S.-H., Park, T., Park, H.D., Jung, D.-H., Kim, J.Y., 2019. Development of Wireless Sensor Node and Controller Complying with Communication Interface Standard for Smart Farming. J. Biosyst. Eng. 23, 41–45.

Park, S., Im, J., Jang, E., Rhee, J., 2016. Drought assessment and monitoring through blending of multi-sensor indices using machine learning approaches for different climate regions. Agric. For. Meteorol. 216, 157–169.

Patil, S. L., Tantau, H.J., Salokhe, V.M., 2008. Modelling of tropical greenhouse temperature by auto regressive and neural network models. Biosyst. Eng. 99, 423–431. https://doi.org/10.1016/j.biosystemseng.2007.11.009

Patil, S L, Tantau, H.J., Salokhe, V.M., 2008. Modelling of tropical greenhouse temperature by auto regressive and neural network models. Biosyst. Eng. 99, 423–431. https://doi.org/http://dx.doi.org/10.1016/j.biosystemseng.2007.11.009

Pawlowski, A., Beschi, M., Guzmán, J.L., Visioli, A., Berenguel, M., Dormido, S., 2016. Application of SSOD-PI and PI-SSOD event-based controllers to greenhouse climatic control. ISA Trans. 65, 525–536. https://doi.org/10.1016/j.isatra.2016.08.008

Pawlowski, A., Sánchez-Molina, J.A., Guzmán, J.L., Rodríguez, F., Dormido, S., 2017. Evaluation of event-based irrigation system control scheme for tomato crops in greenhouses. Agric. Water Manag. 183, 16–25. https://doi.org/https://doi.org/10.1016/j.agwat.2016.08.008

Piñón, S., Camacho, E.F., Kuchen, B., Peña, M., 2005. Constrained predictive control of a greenhouse. Comput. Electron. Agric. 49, 317–329.

Puterman, M.L., 2014. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural networks 12, 145–151.

Qian, T., Dieleman, J.A., Elings, A., De Gelder, A., Marcelis, L.F.M., 2015. Response of tomato crop growth and development to a vertical temperature gradient in a semi-closed greenhouse. J. Hortic. Sci. Biotechnol. 90, 578–584. https://doi.org/10.1080/14620316.2015.11668717

Ramos Ruiz, G., Lucas Segarra, E., Fernández Bandera, C., 2019. Model predictive control optimization via genetic algorithm using a detailed building energy model. Energies 12, 34.

Rodríguez, F., Berenguel, M., Guzmán, J.L., Ramírez-Arias, A., 2015. Modeling and Control of Greenhouse Crop Growth. https://doi.org/10.1007/978-3-319-11134-6

Rouphael, Y., Raimondi, G., Caputo, R., Pascale, S. De, 2016. Fertigation Strategies for Improving Water Use Efficiency and Limiting Nutrient Loss in Soilless *Hippeastrum* production. HortScience 51, 684–689.

Roy, J.C., Boulard, T., Kittas, C., Wang, S., 2002. PA—Precision Agriculture. Biosyst. Eng. 83, 1–20. https://doi.org/http://dx.doi.org/10.1006/bioe.2002.0107

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv Prepr. arXiv1609.04747.

Sak, H., Senior, A., Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Fifteenth Annual Conference of the International Speech Communication Association.

Sánchez, J.A., Rodríguez, F., Guzmán, J.L., Arahal, M.R., 2012. Virtual sensors for designing irrigation controllers in greenhouses. Sensors 12, 15244–15266.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117. https://doi.org/https://doi.org/10.1016/j.neunet.2014.09.003

Seginer, I., McClendon, R.W., 1992. Methods for optimal control of the greenhouse environment. Trans. ASAE.

Shamshiri, R.R., Bojic, I., van Henten, E., Balasundram, S.K., Dworak, V., Sultan, M., Weltzien, C., 2020. Model-based evaluation of greenhouse microclimate using IoT-Sensor data fusion for energy efficient crop production. J. Clean. Prod. 121303.

Shen, X., Zhang, G., Wu, W., Bjerg, B., 2013. Model-based control of natural ventilation in dairy buildings. Comput. Electron. Agric. 94, 47–57. https://doi.org/10.1016/j.compag.2013.02.007

Shin, H.-H., Cho, G.-D., Woo, S.-G., 2015. Analysis of Manpower Employment Demand Based on Agricultural Forms in Accordance of the Aging South Korean Agricultural Managers. Korean Soc. Int. Agric. 27, 138–148.

Shin, J.H., Park, J.S., Son, J.E., 2014. Estimating the actual transpiration rate with compensated levels of accumulated radiation for the efficient irrigation of soilless cultures of paprika plants. Agric. Water Manag. 135, 9–18. https://doi.org/https://doi.org/10.1016/j.agwat.2013.12.009

Shindel, P., Yadav, S., Rudrake, S., Kumbhar, P., 2019. Smart traffic control system using YOLO. Int. Res. J. Eng. Technol.(IRJET) 6, 966–970.

Signore, A., Serio, F., Santamaria, P., 2016. A targeted management of the nutrient solution in a soilless tomato crop according to plant needs. Front. Plant Sci. 7, 391.

Soldatos, A.G., Arvanitis, K.G., Daskalov, P.I., Pasgianos, G.D., Sigrimis, N.A., 2005. Nonlinear robust temperature–humidity control in livestock buildings. Comput. Electron. Agric. 49, 357–376. https://doi.org/http://dx.doi.org/10.1016/j.compag.2005.08.008

Somov, A., Shadrin, D., Fastovets, I., Nikitin, A., Matveev, S., seledets, I., Hrinchuk, O., 2018. Pervasive Agriculture: IoT-Enabled Greenhouse for Plant Growth Control. IEEE Pervasive Comput. 17, 65–75. https://doi.org/10.1109/MPRV.2018.2873849

Stanghellini, C., 1992. Environmental control of greenhouse crop transpiration. J. Agric. Eng. Res. 51, 297–311.

Stanghellini, C., 1987. Transpiration of greenhouse crops: an aid to climate management.

Sukhatme, G.S., Dhariwal, A., Zhang, B., Oberg, C., Stauffer, B., Caron, D.A., 2007. Design and development of a wireless robotic networked aquatic microbial observing system. Environ. Eng. Sci. 24, 205–215.

Taki, M., Abdanan Mehdizadeh, S., Rohani, A., Rahnama, M., Rahmati-Joneidabad, M., 2018. Applied machine learning in greenhouse simulation; new application and analysis. Inf. Process. Agric. 5, 253–268. https://doi.org/10.1016/j.inpa.2018.01.003

Taki, M., Ajabshirchi, Y., Ranjbar, S.F., Rohani, A., Matloobi, M., 2016. Heat transfer and MLP neural network models to predict inside environment variables and energy lost in a semi-solar greenhouse. Energy Build. 110, 314–329. https://doi.org/https://doi.org/10.1016/j.enbuild.2015.11.010

Tarange, P.H., Mevekari, R.G., Shinde, P.A., 2015. Web based automatic irrigation system using wireless sensor network and embedded Linux board, in: 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]. pp. 1–5. https://doi.org/10.1109/ICCPCT.2015.7159327

Tchamitchian, M., Kittas, C., Bartzanas, T., Lykas, C., 2005. Daily temperature optimisation in greenhouse by reinforcement learning, in: 16th IFAC World Congress, Prague, Czech Republic.

Teófilo, R.F., Martins, J.P.A., Ferreira, M.M.C., 2009. Sorting variables by using informative vectors as a strategy for feature selection in multivariate regression. J. Chemom. 23, 32–48. https://doi.org/10.1002/cem.1192

Theodorakopoulos, A., Rovithakis, G.A., 2014. A simplified adaptive neural network prescribed performance controller for uncertain MIMO feedback linearizable systems. IEEE Trans. neural networks Learn. Syst. 26, 589–600.

Theodoridis, S., 2015. Chapter 5 - Stochastic Gradient Descent: The LMS Algorithm and its Family, in: Theodoridis, S.B.T.-M.L. (Ed.), . Academic Press, Oxford, pp. 161–231. https://doi.org/https://doi.org/10.1016/B978-0-12-801522-3.00005-7

Vadiee, A., Martin, V., 2012. Energy management in horticultural applications through the closed greenhouse concept, state of the art. Renew. Sustain. Energy Rev. 16, 5087–5100. https://doi.org/https://doi.org/10.1016/j.rser.2012.04.022

van Straten, G., Challa, H., Buwalda, F., 2000. Towards user accepted optimal control of greenhouse climate. Comput. Electron. Agric. 26, 221–238. https://doi.org/https://doi.org/10.1016/S0168-1699(00)00077-6

Villarreal-Guerrero, F, Kacira, M., Fitz-Rodríguez, E., Kubota, C., Giacomelli, G.A., Linker, R., Arbel, A., 2012. Comparison of three evapotranspiration models for a greenhouse cooling strategy with natural ventilation and variable high pressure fogging. Sci. Hortic. (Amsterdam). 134, 210–221. https://doi.org/http://dx.doi.org/10.1016/j.scienta.2011.10.016

Villarreal-Guerrero, Federico, Kacira, M., Fitz-Rodríguez, E., Linker, R., Kubota, C., Giacomelli, G.A., Arbel, A., 2012. Simulated performance of a greenhouse cooling control strategy with natural ventilation and fog cooling. Biosyst. Eng. 111, 217–228. https://doi.org/http://dx.doi.org/10.1016/j.biosystemseng.2011.11.015

Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J.P., Jaderberg, M., Vezhnevets, A.S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T.L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., Silver, D., 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575, 350–354. https://doi.org/10.1038/s41586-019-1724-z

Wang, H., Sánchez-Molina, J.A., Li, M., Berenguel, M., Yang, X.T., Bienvenido, J.F., 2017. Leaf area index estimation for a greenhouse transpiration model using external climate conditions based on genetics algorithms, back-propagation neural networks and nonlinear autoregressive exogenous models. Agric. Water Manag. 183, 107–115. https://doi.org/http://dx.doi.org/10.1016/j.agwat.2016.11.021

Wang, W., Xu, L., Hu, H., 2015. Neuron adaptive PID control for greenhouse environment. J. Ind. Prod. Eng. 32, 291–297. https://doi.org/10.1080/21681015.2015.1048752

Whitley, R., Medlyn, B., Zeppel, M., Macinnis-Ng, C., Eamus, D., 2009. Comparing the Penman–Monteith equation and a modified Jarvis–Stewart model with an artificial neural network to estimate stand-scale transpiration and canopy conductance. J. Hydrol. 373, 256–266. https://doi.org/https://doi.org/10.1016/j.jhydrol.2009.04.036

Wilks, Y., 2019. Artificial Intelligence: Modern Magic Or Dangerous Future? Icon Books.

Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B., 2017. The marginal value of adaptive gradient methods in machine learning, in: Advances in Neural Information Processing Systems. pp. 4148–4158.

Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.-J., 2017. Big Data in Smart Farming – A review. Agric. Syst. 153, 69–80. https://doi.org/10.1016/J.AGSY.2017.01.023

Xu, D., Du, S., van Willigenburg, L.G., 2018. Optimal control of Chinese solar greenhouse cultivation. Biosyst. Eng. 171, 205–219. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2018.05.002

Yang, Xincong, Li, H., Yu, Y., Luo, X., Huang, T., Yang, Xu, 2018. Automatic pixel-level crack detection and measurement using fully convolutional network. Comput. Civ. Infrastruct. Eng. 33, 1090–1109.

Yano, A., Tsuchiya, K., Nishi, K., Moriyama, T., Ide, O., 2007. Development of a Greenhouse Side-ventilation Controller driven by Photovoltaic Energy. Biosyst. Eng. 96, 633–641. https://doi.org/https://doi.org/10.1016/j.biosystemseng.2006.12.012

Yeo, U., Lee, I., Kwon, K., Ha, T., Park, S., Kim, R., Lee, S., 2016. Analysis of Research Trend and Core TechnologiesBased on ICT to Materialize Smart-farm. Prot. Hortic. Plant Fact. 25, 30–41.

Yoo, S.-H., Lee, S.-H., Choi, J.-Y., Im, J.-B., 2016. Estimation of potential water requirements using water footprint for the target of food self-sufficiency in South Korea. Paddy water Environ. 14, 259–269.

Yu, H., Chen, Y., Hassan, S.G., Li, D., 2016a. Prediction of the temperature in a Chinese solar greenhouse based on LSSVM optimized by improved PSO. Comput. Electron. Agric. 122, 94–102. https://doi.org/10.1016/j.compag.2016.01.019

Yu, H., Chen, Y., Hassan, S.G., Li, D., 2016b. Prediction of the temperature in a Chinese solar greenhouse based on LSSVM optimized by improved PSO. Comput. Electron. Agric. 122, 94–102. https://doi.org/http://dx.doi.org/10.1016/j.compag.2016.01.019

Zeiler, M.D., 2012. Adadelta: an adaptive learning rate method. arXiv Prepr. arXiv1212.5701.

Zeng, S., Hu, H., Xu, L., Li, G., 2012. Nonlinear Adaptive PID Control for Greenhouse Environment Based on RBF Network. Sensors 12, 5328–5348. https://doi.org/10.3390/s120505328

Zhang, Y., Wang, J., 2001. Recurrent neural networks for nonlinear output regulation. Automatica 37, 1161–1173. https://doi.org/https://doi.org/10.1016/S0005-1098(01)00092-9

Huh, M,Y et al., 2018. RS485 MODBUS interface between greenhouse controller and nutrient supplynode in smart greenhouse, Telecommunications Technology Association, TTA

국문 초록 (Korean)

시설재배는 노지재배와 다르게 재배환경을 조절할 수 있어서 기상환경 조건, 작물 재배 조건 등 재배환경의 분석을 통해 생산성의 증대와 품질 향상을 달성할 수 있다. 이 때문에 시설재배에서 정밀제어와 냉·난방을 활용하는 온실이 증가하고 있다. 초기의 시설재배는 간단한 농업용수의 공급 정도였으나 기술의 발달과 고도화로 인하여 작물 재배과정 전반에 걸쳐 자동화가 이루어지고 있다. 스마트팜 기술은 첨단 ICT, 빅데이터, 자동화/로봇 기술을 이용하여 농민의 개입을 최소화하면서 생산성과 품질을 극대화하고 수익 창출형 모델을 확립이 가능한 신개념의 농법이다. 온실의 복합 환경조절은 특히 농작물의 수확량과 품질에 직접적인 영향을 주는 핵심적인 기술 요소로 국내 선진화된 스마트팜 농가는 대부분이 해외 복합 환경제어 시스템을 사용하고 있다. 하지만 해외 제어시스템에서는 국내의 기후조건 국내산 작물 품종의 재배환경 등을 고려하고 있지 않으며, 농민들이 직접 제어 설정 값을 미세 조정해야 하는 어려움을 겪고 있다. 이러한 원인으로, 농민들에게 전문적인 제어 소프트웨어 교육 훈련 및 숙달이 필수적이고, 작동 숙달 미숙으로 인해 제어 효율이 기대치만큼 못 미치고 있다. 궁극적으로, 환경제어를 위한 선형제어 모델은 비선형성이 크며 매우 동적인 온실 환경에 적용하기에는 한계가 있다. 따라서, 스마트팜 농가의 환경 빅데이터를 활용한 인공지능 기반의 정밀한 제어 시스템을 개발하는 것이 본 연구의 목적이다.

본 연구에서는 온실 내부 대기환경인 온도, 습도 및 CO_2 농도를 관리하기 위해 외부 영향인자가 고려된 선형 알고리즘 (PD-band, P-band) 기반 제어 방법을 제안하였고 그 성능을 검증하였다. 구동기별 설정인자에 대한 설정 (영향 계수)을 최적화하기 위해 반응 표면 분석 방법을 실험설계를 통해 수행하였다. 결과를 바탕으로 최적의 환기 제어 조건을 조사하였으며, 각 요인에 대한 영향 값을 실제 온실 온도 제어에 적용하였고 제어 성능을 평가하였으며, 그 결과 1.25 ℃의 RMSE 값은 최적화 된 계수로 인해 개선된 성능을 확인 하였다.

온실의 환경변화를 예측하기 위해서 딥러닝 알고리즘을 적용하여 데이터 기반의 모델링을 수행하였다. ANN, NARX 및 RNN-LSTM 모델은 온실 작물의 성장에 직접적인 영향을 미치는 온도, 습도 및 CO_2 농도 변화 예측 성능을 비교하였다.

RNN-LSTM 모델은 각각 5 % 이하의 SEP 와 R2 0.81-0.96 에서 세 가지 환경인자 예측성능을 보였다. 또한 다양한 훈련 조건을 비교하기 위해 5-30 분의 타임스텝에 대한 예측 성능을 비교하였다. 추가적으로, CNN-LSTM 을 활용하여 온실 환경변화 예측에 적용하였는데, 입력 인자들의 정보를 시간열에따라 2 차원화하여 예측모델을 설계하였다. 개발된 CNN-LSTM 기반의 환경 예측 모델은 RNN-LSTM 모델과 비교하여 나은 성능을 확인 하였다. 최종적으로는, 이러한 모델들을 소형 임베디드보드에 탑재하여, 자동으로 온실의 데이터를 하루 간격으로 수집하여 매일 학습 및 모델 수정이 되는 시스템을 제안하여, 다양한 구조의 온실에 적용할 때 초기 학습이 용이하도록 하였다. 이 연구의 결과는 온실 제어에서 딥러닝 기반 예측 모델의 적용 가능성 및 환경해석에 도움이 될 것으로 판단된다.

Output feedback neural network 구조를 기반으로 하는 예측모델의 최적 해를 추종하는 제어 방법을 제안하여, 온실의 30 분 후에 발생하는 기후 변화로부터 목표 설정 값과의 비용을 계산하고 이를 최적화된 제어 구동기의 신호를 결정하였다. 인공지능에 활용되는 SGD, Rmsprops, Adadelta 등 다양한 최적화 기법들을 비교하였으며, 시뮬레이션을 통해 제어 가능성을 확인 하였다. 또한, 현장 실험을 통해 제안된 방법으로 환경제어 신호를 결정할 경우 기존 선형 알고리즘보다 환경 변화에 더 유리하다는 다중 창 환기 제어 실험을 통해 검증하였다. 에너지 절약 효과를 위해, 액추에이터의 동작에 소비되는 에너지를 상대적으로 비교하여 Cost gate 를 활용하여 최적화를 위한 함수에 부분 제한 변수로 사용이 가능함을 제안하였고, 시뮬레이션과 실제 현장 적용을 통해 실제 에너지 절약 효과를 확인하였다. 시뮬레이션 결과 여름철 기후에서는 난방 신호를 결정하거나 유동 팬 동작을 기피하는 인공지능 모델 추론을 확인 하였으며, 이는 앞으로 인공지능의 활용에 있어서 제한된 동작범위와 현실적은 목표 설정이 요구됨을 확인 할 수 있었다. 이러한 점을 개선한다면 최종적으로 이를 통해 입력 에너지를 절약하고 경제적 기여를 할 수 있으며 가까운 시일 내에 다양한 분야에서 응용이 가능할 것으로 판단된다.

8. APPENDICES

8.1. DESIGN AND APPLICABILITY OF REINFORCEMENT LEARNING TO THE OPTIMAL CLIMATE CONTROL

8.1.1. THEORY OF REINFORCEMENT LEARNING (RL)

Reinforcement learning (RL) is the method of determining an action in a given situation so to maximize a reward. Most of the machine learning techniques give instructions directly to the learning agents about what actions to take to achieve given goals. On the other hand, RL does not designate any action to the learning agents. Each agent finds out an action that could maximize a reward for each action critic. The action thus taken has a direct effect on both the reward and subsequent situation. In other words, all the subsequent rewards are affected. RL can be understood only when the definition of a learning problem is distinguished from that of a learning technique. Any technique that helps in solving a problem can be regarded as an RL technique. In comparison with other learning techniques, RL is clearly distinguished with respect to trial and error learning and delayed reward. Every agent of RL is trained on the correlation between an action and a result through the two characteristics.

Most current studies focus on supervized learning, which seems to be similar to RL, except for two techniques that are fundamentally different. Supervised learning (SL) utilizes samples with given labels. Although this method is very useful, it is not appropriate for an interaction-based learning technique. In a case wherein a machine is made to investigate a new topography, SL is usually not applicable.

The basic element of SL is interaction between an agent and the environment, and in which the following factors are also included. The agent of SL is usually trained in the following procedure. Some states concerning the agent and environment are defined. At a specific point, the agent receives a state containing information about the surrounding environment (Figure 143).

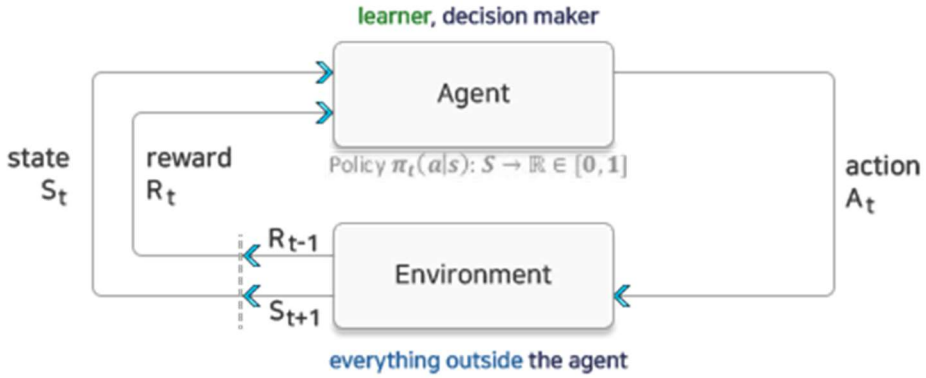

Figure 143. Basic components of supervised reinforcement learning

The policy about actions, which the agent is expected to take, is defined. This policy executes the decision-making function. The agent's action for an input state is determined by the policy. The agent acts according to the decision of the previous step. The environment reacts to the agent's action, and the agent receives a reward from the environment. The agent records information (type and state) about the reward. The RL system implements multiple tasks simultaneously. Trial and learning is executed, the model of the environment to which the agent belongs is trained, and the next action is determined using the developed model.

8.1.2. APPLICATION OF RL IN THE CONTROL OF A NUTRIENT SOLUTION SUPPLY IN A SUBSTRATE

To employ RL in the control of a nutrient solution supply in a substrate, an agent-environment structure must be constructed. In this study, a Q-learning environment was built, which combined the critic and actor methods. The amount of the supply, concentration, and time of nutrient solution were determined, and a reward was given in the form of an error from the state of the root zone.

An actor-critic policy gradient-based RL agent with an artificial neural network was implemented in the fertigation control system. This deep RL algorithm finds how to maximize future rewards, thus the control performance is maximized (Ban and Kim, 2017).

In this study, we focus on RL, which is an area of machine learning. Its main idea includes having an agent with the possibly zero knowledge of the structure of the

surrounding environment, except for its observations (states). One can train this agent to perform optimal actions for the given state, where each action is rewarded with some values. The agent learns the optimal behavior (policy) by trial and error to maximize the cumulative gain. For plant growth, positive rewards are given for good growth rates and final crop yield, whereas, negative rewards are given for the consumption of resources (Somov et al., 2018).

The mathematical model for RL is the Markov decision process. It is defined as a tuple with the state space of environment (air, temperature and humidity, solution content, and plant type and size), the set of actions the agent can take (turn on/off light, irrigate, and harvest), and a reward function that returns the immediate reward received by the agent upon taking an action. The Markov property implies that the state transition depends only on the current state and action taken by the agent. The policy function describes the behavior of the agent, i.e., which action it takes while being in the state.

The output corresponds to an action to be applied to the environment, and the inputs to the actor are based only on observation. The network parameters are updated with critic output value. The critic network has a 3-layered FNN model, which has a fuse operation on the second layer (Figure 144). Another single layer receives the output of the actor to be merged with the encoded observation information. The backpropagation of this critic network follows both subnetworks at the same time; hence, the network naturally shows how to optimize the actor policy by evaluating it with respect to a simultaneous observation. Network parameters are updated to reduce the mean squared error between the previously predicted Q value and discounted real rewards (Ban and Kim, 2017; Tchamitchian et al., 2005).

.

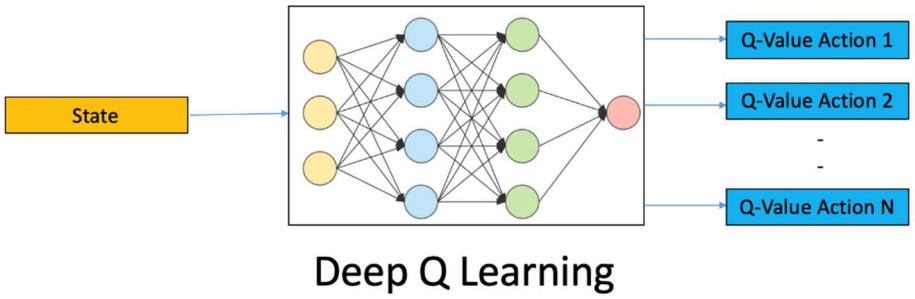

Figure 144. Network architecture of the RL agent consists of multi-independent deep neural networks.

8.1.3. REINFORCEMENT LEARNING BASED PREDICTIVE CLIMATE CONTROL SYSTEM

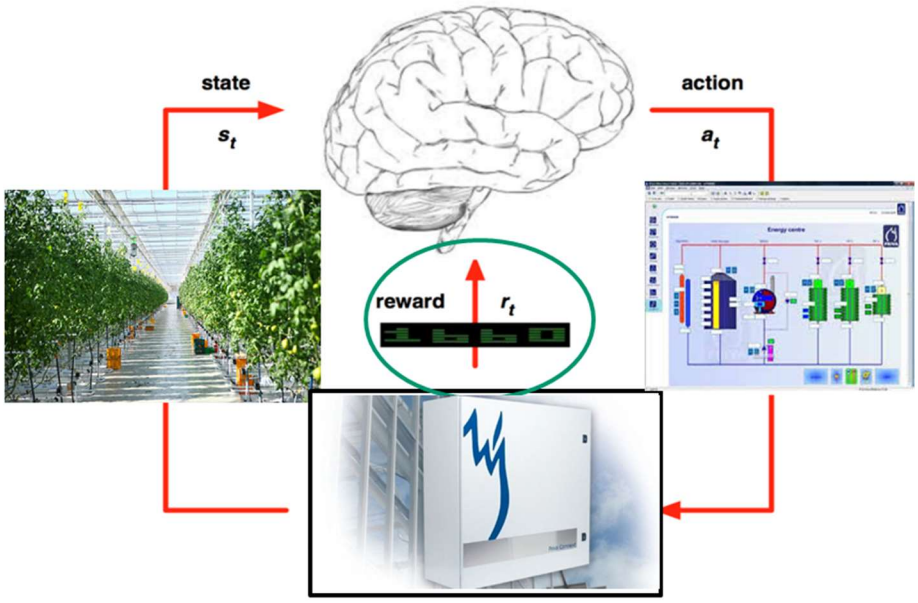

Figure 145. Concept of RL control of irrigation for greenhouse rootzone environmental control.

RL has a potential application in the agricultural sector because there is no existing control model for controlling the orientation of crops. For this reason, related studies on the modeling of various physiological and growth phenomena of crops are underway. Therefore, the application of RL in the control of the greenhouse environment would be a considerably meaningful approach.

We designed the algorithm by selecting the climate control system as the most

suitable environmental control for RL. Figure 105 shows a concept that combines RL concepts with climate control. The trigger on/off and up/down switchs, which control ventilation rate (% cooling and heating) are also considered, with +/− switches for adjusting the rate (%). If the RL model based on climate control is conducted without the preceding model, the control method based on the preceding model is devised because it is controlled to the extent that it adversely affects the crop. The preceding model was derived from the CNN-LSTM environmental prediction model and optimizer. Based on these, we designed the final control signal decision by summing the tuned control history value of RL and control history of the preceding model at the ratio of about 1:1 (it was not fixed).

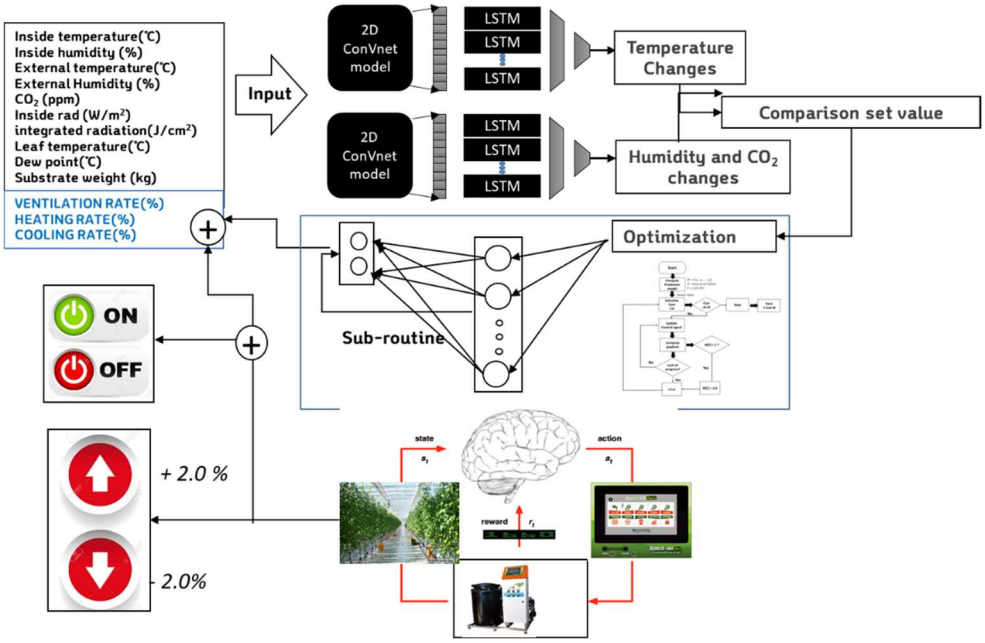

Figure 146. Control nodes based on RL and generator neural network control

8.1.4. REWARDS DESIGN FOR IMPLEMENTING REINFORCEMENT LEARNING

The agent used could be the irrigation controller, state is the greenhouse environment, and supply of nutrient solution must be the action. Two reward functions were selected, including the evapotranspiration rate of the crop and the rate of drainage solution. The reward was designed to create higher scores as the evapotranspiration increases and the amount of discharged solution decreases. This

is calculated by applying a bandwidth of about 10% to the average level of the average historical data. Both the parameters store the accumulated amount using a load cell and a flowmeter, as shown in Figure 148.

The most important factor for the application of RL in nutrient control is the design of rewards. RL proceeds in the direction in which the most statistical compensation is obtained for actions to be taken in the environment. Therefore, to have an existing gap of current irrigation technologies, it is necessary to set the concept of reward for the growth and physiology of the crops rather than a simple set-point-based control. Therefore, this chapter introduces RL design for a novel control method that operates in a such away that encourages the evapotranspiration of crops and energy-saving. **Error! Reference source not found.** and Figure 148 show two guidelines for compensation, which are designed to be used in RL by calculating their relative reward values.

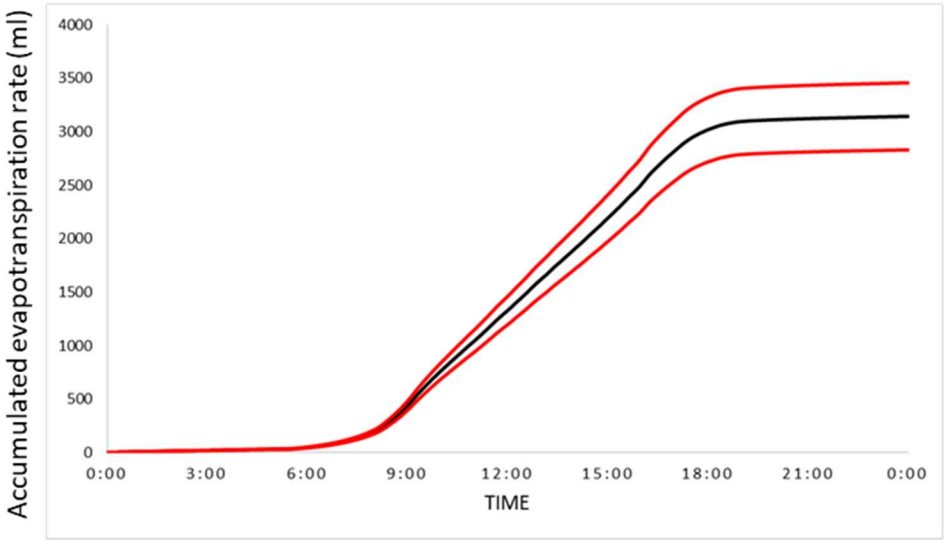

Figure 147. Example of a reward scoreline for ET input rate during a day.

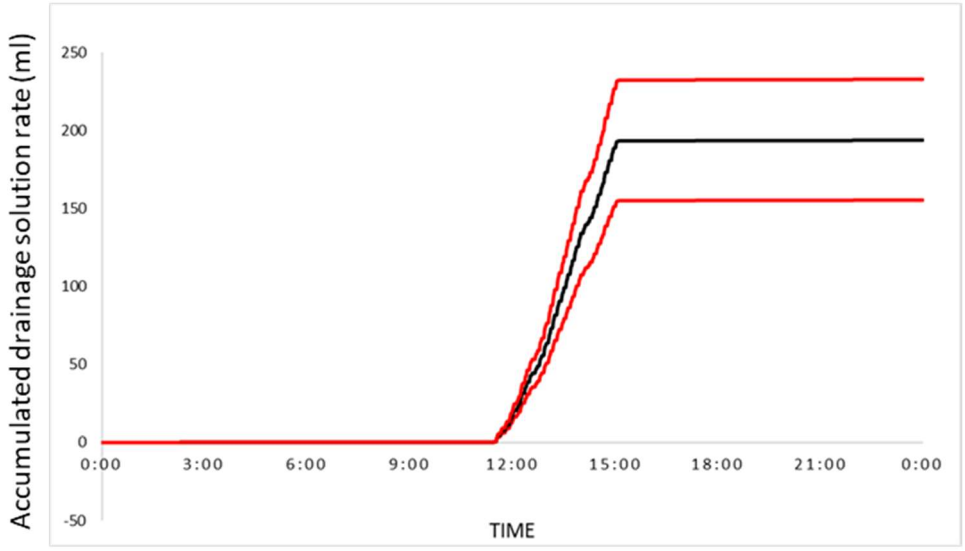

Figure 148. Example of a reward scoreline for energy input rate during a day.

Algorithm flow for Q-learning in Irrigation

```
    Initialize Q(s , a) arbitrarily
2   repeat
        for all e(s , a) do
4           e(s , a) <-- 0
        end for
6       for   i =1, i++, while   i <n+1 do
            s ← State (I , Rei);
8           Get action a
            [Rei +1, Ii   ] ← SimRE ( I , a)
10          S' ← State(I, Rei)
            a' ← greedy (e, Q, Si+1)
12          if   I < n then
                r ← 0
14          else
                E ← Sum(RE1) * EvapoCost
16              D ← Sum(RE2) * DrainCost
                Reward ← E-D
18              if Reward < threshold then
                    r ← -10
```

```
20                  else
                        r ← Reward
22                  end if
                end if
24              delta ← r +rQ(s’ , a’) □Q(s,a)
                e(s , a) ← e(s , a) +1
26              for all s and a do
                Q(s,a) ← Q(s,a) +αδe(s,a)
28              e(s,a) ← λe(s,a)
                end for
30               a ← a’
            end for
32      until Q(s,a) coverages / policy sufficiently stabilized
```

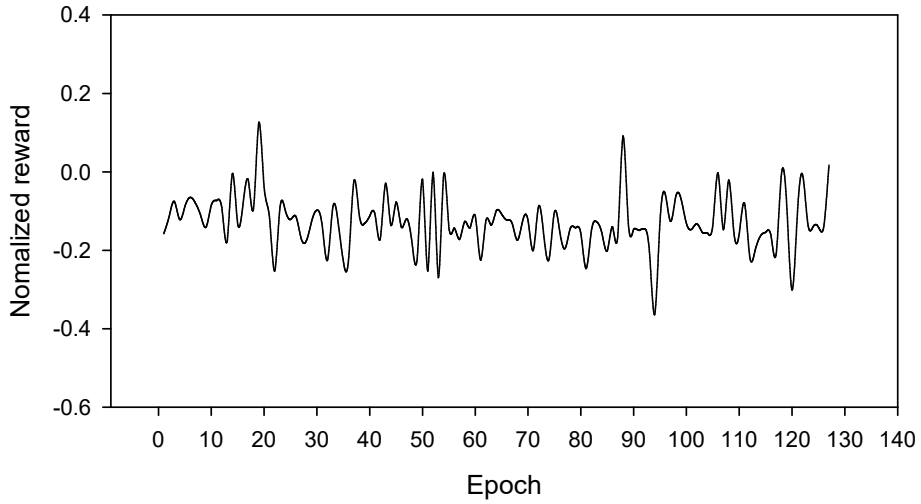

Figure 149. Reward simulation obtained through reinforced learning-based environmental control.

- **CODES**

The core code used in this study will be released through the online open platform Github : jeoguss/Deep-learning-GH-control.git