A Study on the Prediction of Horticultural Paprika Crop Growth in Artificial Intelligence Infrastructure

1st GwangHoon Jung
dept. Information and Communication Engineering
Sunchon National University
Suncheon si, Jeollanam-do
gwanghoon5035@gmail.com

2nd Meong Hun Lee
dept. Convergence Biosystems Mechanical Engineering
Sunchon National University
Suncheon si, Jeollanam-do
leemh777@scnu.ac.kr

3rd Hyun Yoe*
dept. Information and Communication Engineering
Sunchon National University
Suncheon si, Jeollanam-do
yhyun@scnu.ac.kr

*Corresponding author

***Abstract*— Smart farms are gaining significant attention as a solution to the sustainability crisis in rural areas. Challenges such as stagnant income, exports, and growth rates, driven by the aging population in the agricultural and livestock industries, a shortage of young successors to farming, decreasing production areas, and declining investments, are compounded by the increasing instability in crop production and losses caused by ongoing climate change. This study focuses on developing an artificial intelligence-based horticultural paprika crop growth prediction system tailored to environmental conditions such as temperature, humidity, and wind direction. Data were collected from facility horticultural farms using weather, environmental, and specialized sensors, with the measurements stored in a database. Preprocessing was performed on the stored data, and since the dataset consisted of time-series data, time periodicity was incorporated using trigonometric functions. The dataset was split into training, validation, and test sets, and data scaling was applied to normalize the feature ranges for compatibility with AI models. Various models, including Random Forest, Support Vector Machine (SVM), Boosting algorithms, and time series models, were trained to predict crop growth based on external environmental factors (temperature, humidity, wind direction, wind speed, and sunlight) and the actual versus predicted values. The model with the highest accuracy was selected for further analysis. The results of this research demonstrate the potential to increase crop productivity and yields by creating optimal growth conditions. Additionally, the study contributes to cost reduction and environmental sustainability by optimizing the use of pesticides and fertilizers. Beyond addressing environmental factors, the system leverages historical data to predict the likelihood of disease outbreaks and pest infestations. This approach not only improves farm yields but also addresses workforce challenges in the agricultural and livestock industries, offering solutions to the aging population and the shortage of young successors in farming.**

Keywords*— *Paprika Environmental Forecast, smart greenhouse, RandomForest, linear regression, Grid Search, Bayesian optimization, support vector, Hyperparameter tuning, precision agriculture

I. INTRODUCTION

Smart farms are emerging as a promising solution to address the sustainability challenges faced by rural areas. These challenges include stagnant income, limited export and growth rates due to an aging population in the agricultural and livestock industries, a shortage of young successors to farming, decreasing production areas, and declining investments [1]. Smart farms integrate facility horticulture with advanced technologies such as IoT, big data, AI, automation systems, and robotics to remotely or automatically manage the growing environment for crops. Presently, the adoption of AI in smart farms highlights the need for an enhanced integrated environmental control system capable of upgrading operational efficiency [2].

Crops are highly influenced by environmental factors during their growth, necessitating precise environmental control to optimize yield. Among various factors, temperature and humidity are crucial for crop development. Temperature regulates key processes such as germination, growth duration, differentiation, and flowering. Deviations from the optimal growth temperature can inhibit growth and, in severe cases, lead to crop failure [3].

Similarly, humidity plays a vital role in plant physiology. Inappropriate humidity levels can cause disorders such as fruit drop, leaf wilting, and stomatal closure, while also increasing the likelihood of pests and diseases. Improper control of these environmental factors can significantly deteriorate the growth environment, reduce production, and exacerbate pest-related issues [4].

In facility-based horticulture, devices are available to control internal temperature and humidity. Since these parameters vary based on actions such as opening or closing switches, predicting changes in temperature and humidity can serve as an effective strategy for precise environmental control [5].

This study collected environmental variables essential for crop growth in facility horticulture and employed time series analysis to predict growth environments. The performance of the random forest model was evaluated using OOB (Out-Of-Bag) scores, and hyperparameter tuning was conducted through grid search and Bayesian optimization to enhance model performance. The RMSE value was calculated as part of the optimization process. In addition to random forest, the model was also assessed using support vector machines (SVM) and Gradient Boosting.

Furthermore, an AI-based field crop growth information prediction system utilizing the random forest model was implemented on actual farms to provide one-month data forecasts. Applying these research findings to open-field farms could help address constantly changing external factors, thereby optimizing crop management and yield planning. This approach is anticipated to enable more efficient operations, including resource management, market demand forecasting, and price management.

II. Related Research

A. Introduction of RandomForest

Random Forest is a machine learning algorithm built on an ensemble of multiple decision trees. Each tree in the model is independently trained on a random subset of the data, and the algorithm combines the predictions of these individual trees to produce a final decision .

The trees are constructed using bootstrap sampling, a process in which samples are randomly selected from the entire dataset with replacement. This method increases the diversity of the model by reducing the correlation between trees. Furthermore, instead of using all features during the splitting process, each tree considers only a randomly selected subset of features to determine the splits. This approach allows Random Forest to capture different aspects of the data, enhancing the model's generalization capabilities.

An additional strength of Random Forest is its ability to assess the importance of each feature, enabling the identification of variables that significantly influence predictions. One of its primary advantages is its resilience to overfitting, as the ensemble approach mitigates the risk of individual trees overfitting to specific data samples or noise [6].

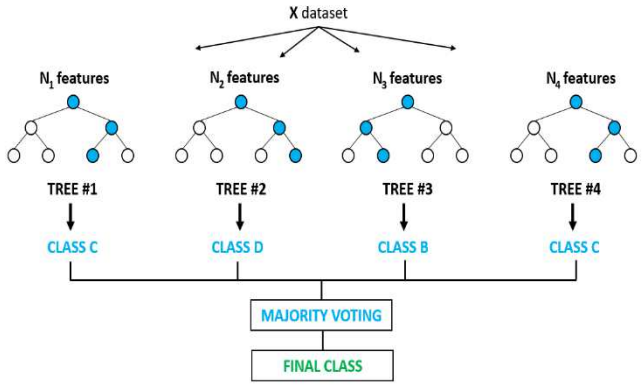

Figure 1. Random Forest Decision Tree

B. Introduction of Hyperparameter tuning

Hyperparameter tuning is a critical process in machine learning that involves optimizing model parameters, such as the learning rate or the number of hidden layers in a neural network, prior to training. These hyperparameters significantly influence the model's operation and performance, making their optimization essential .

Effective hyperparameter optimization directly impacts model efficiency by mitigating the risks of overfitting (when the model is overly complex) and underfitting (when the model is too simple). Various methods can be employed for this process, including grid search, Bayesian optimization, and gradient-based optimization. These techniques use probabilistic models or random searches to explore combinations of hyperparameters, predicting the most effective configurations. Bayesian optimization, in particular, is well-suited for continuous hyperparameters, while gradient-based methods often rely on gradient descent for optimization. Evolutionary algorithms, inspired by natural processes like mutation and crossover, also play a role in identifying optimal hyperparameter sets by balancing model complexity and generalization capabilities.

The effectiveness of hyperparameter tuning is typically assessed through cross-validation, a method where the model is trained and validated on different subsets of the data. Performance metrics tailored to the problem at hand, such as classification or regression, are used to evaluate the model. Commonly used metrics include accuracy, precision, recall, F1 score, and mean squared error.

These metrics provide insight into the model's ability to generalize and perform effectively across unseen data[7].

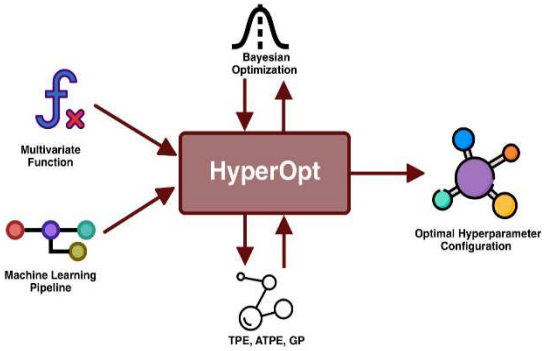

Figure 2. Hyperparameter Tuning Fundamentals

C. Introduction of Grid Search

Grid search is a commonly used technique in machine learning for hyperparameter tuning. Its goal is to systematically explore multiple combinations of hyperparameter options to identify the best configuration for a specific model and dataset. This involves constructing a grid of all possible hyperparameter combinations [8].

The model is trained for each combination within this grid, enabling a thorough and precise search through a predefined subset of the hyperparameter space. This method is highly beneficial as it helps improve model performance by identifying the optimal hyperparameters, which are those that yield the best results. Model performance is typically evaluated using metrics such as accuracy, precision, recall, or other measures suited to the specific application.

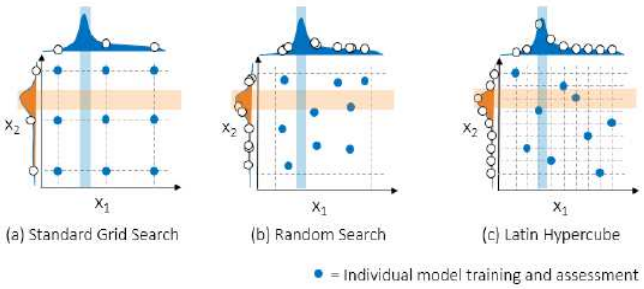

Figure 3. Grid Search Fundamentals

D. Introduction of Bayesian optimization

Bayesian optimization is an advanced method for optimizing objective functions that are costly to evaluate, making it particularly effective for hyperparameter tuning in machine learning models. It is highly efficient at identifying optimal hyperparameters, especially when evaluating the function involves significant computational resources, such as training complex models on large datasets. The fundamental

principle of Bayesian optimization is to use a probabilistic model to approximate the objective function and guide the search for optimal parameters within the hyperparameter space .

Unlike traditional methods like grid search or random search, which do not adapt their strategies based on prior evaluations, Bayesian optimization leverages past results to make informed decisions about where to evaluate next. It approximates the objective function, which is often unknown and expensive to compute, using a stochastic model, typically a Gaussian Process (GP). The GP not only estimates the objective function's values but also provides a measure of uncertainty, offering both a mean estimate and variance for each point in the parameter space. This dual capability makes the Gaussian Process particularly well-suited for Bayesian optimization .

The Bayesian optimization process consists of two key components: a surrogate probability model, which approximates the objective function, and an acquisition function, which determines the next point in the parameter space to evaluate[9].

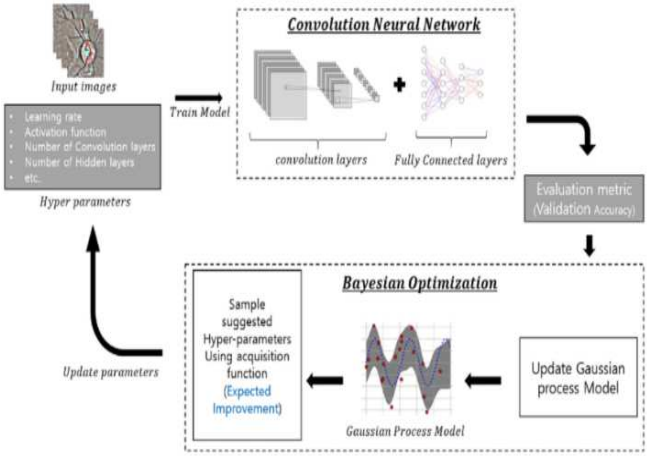

Figure 4 Basic Principles of Bayesian Optimization

E. *Introduction of Support Vector Machine*

Support Vector Machines (SVMs) are a sophisticated class of supervised learning algorithms, primarily designed for classification tasks but also applicable to regression. They are widely recognized for their ability to handle complex datasets and model intricate decision boundaries effectively. The core principle of SVMs is to identify the optimal hyperplane that best separates data classes within the feature space .

In a two-dimensional space, this hyperplane appears as a straight line, but in higher-dimensional spaces (corresponding to the number of features in the data), it becomes a multidimensional surface. The placement of this hyperplane is crucial, as it is determined by the closest data points from each class, known as support vectors. These support vectors are critical in defining the orientation and position of the hyperplane.

SVMs aim to maximize the margin between the hyperplane and the support vectors, where the margin is the distance between the hyperplane and the nearest points from each class. A larger margin typically reduces the generalization error of the model, improving its performance on unseen data .

SVMs are effective for both linear and nonlinear data. When the data is linearly separable, simpler linear SVMs are often used, while more advanced techniques, such as kernel methods, are employed for nonlinear cases to map data into higher-dimensional spaces where linear separation becomes possible[10].

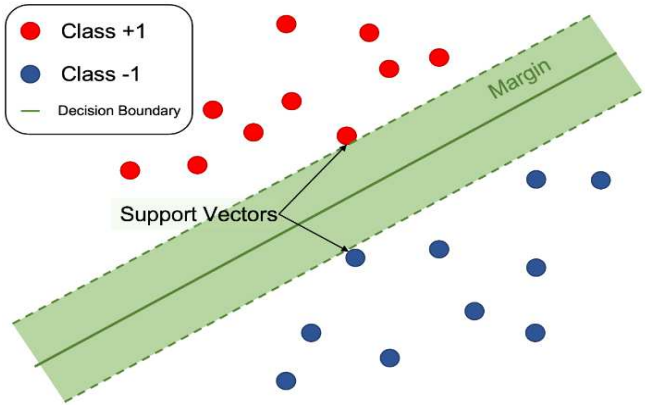

Figure 5 Basic Principles of Support Vector Machine

III. Main Subject

A. *Process of collecting and preprocessing crop growth environment data*

This section outlines the data collection methodology, the characteristics of the collected data, and the data preprocessing steps utilized in this study. Figure 6 presents a photograph illustrating paprika environmental data obtained from actual facility horticulture settings.

Figure 6. Collecting Environmental Data

Paprika data were collected at 5-minute intervals from January 2024 to May 31, 2024. Table 1 summarizes the characteristics of the collected data.

TABLE I. Sensors used and data items collected

	Environmental data information		
	Data	***Datatype***	***Unit***
date	Data collection date	datetime (yyyy-mm-dd)	Day
Temperature	Translation of temperature in 5-minute increments	float64	˚C
Humidity	Translation of humidity in 5 minutes	float64	%
Wind direction	Translation of wind direction every 5 minutes	float64	°

	Environmental data information		
	Data	***Datatype***	***Unit***
Wind speed	Translation of wind speed in 5-minute increments	float64	m/s
Sunlight	amount of sunlight per day	float64	W/m²
Total_ Sunlight	Translation of accumulated sunlight	float64	W/m²

Graph the data to check missing or outlier data.

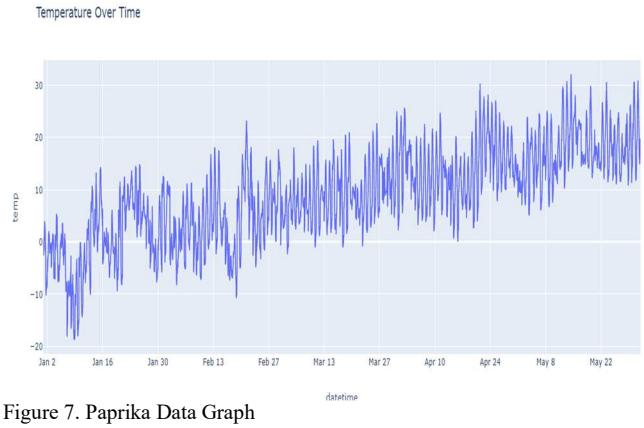

Figure 7. Paprika Data Graph

Looking at the graph, there appear to be no missing values. To confirm this more accurately, we used a code-based method to check for missing values. As a result, it was verified that no missing values were present, as shown in Figure 8.

```
temp          0
humi          0
deg           0
wind_speed    0
sun           0
total_sun     0
dtype: int64
```

Figure 8. Check missing values

Since time series data is heavily influenced by time, we analyzed temperature variations based on the time of day, the daily temperature changes across each day of the week, and the monthly temperature fluctuations.

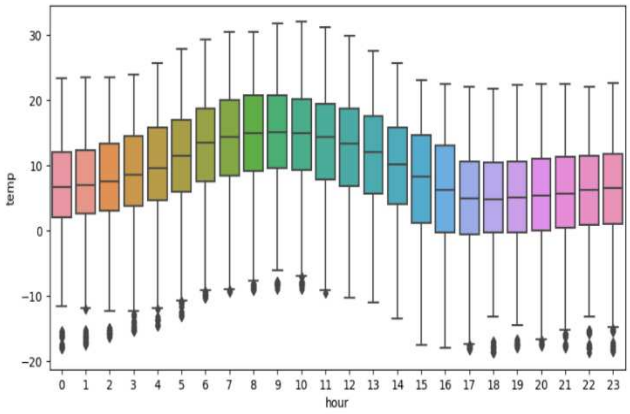

Figure 9. Temperature changes in paprika over time

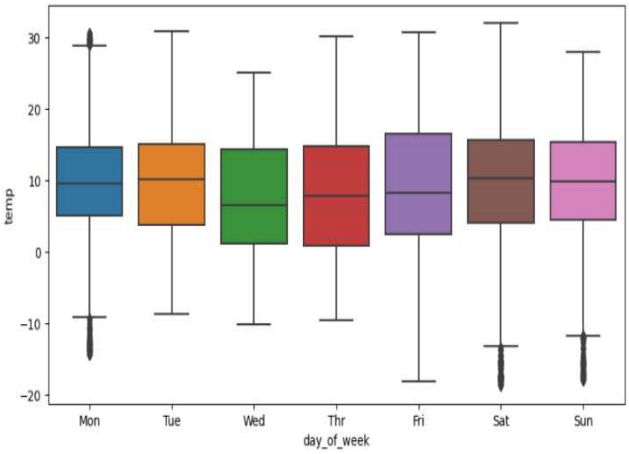

Figure 10. Temperature changes in paprika over the course of a week

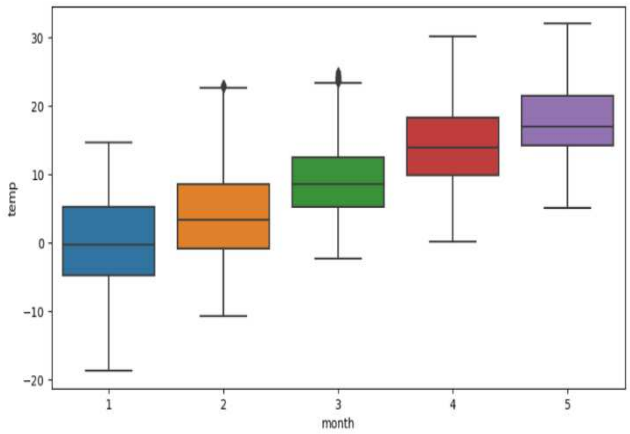

Figure 11. Monthly temperature changes in paprika

B. Preprocessing of crop growth environment datal

The temperature change data over time exhibits daily, weekly, and monthly periodicity, emphasizing the importance of time in predictions. However, since the time resets to 00:00 after every 24 hours, adjustments must be made to account for this periodicity before the data can be utilized effectively.

In this study, sinusoidal functions (sin and cos) were employed to transform the time into a continuous yearly format. Time was converted into seconds to incorporate daily or yearly cycles, with one day expressed as seconds and scaled to day(365.2425)×day to account for leap years within a 365-day year.

After completing basic data preprocessing, the dataset must be divided. It is split into training, validation, and test sets with a ratio of 8:1:1.

After creating a dataset, it is necessary to scale the data. This is because many machine learning and deep learning algorithms assume that all features are centered around 0 and have similar distributions.

C. Results Analysis

This chapter focuses on the process of comparing actual values with predicted values using preprocessed data and machine learning algorithms. The study involved evaluating the performance of random forest, support vector machine, boosting algorithms, and time series analysis to compare actual and predicted values.

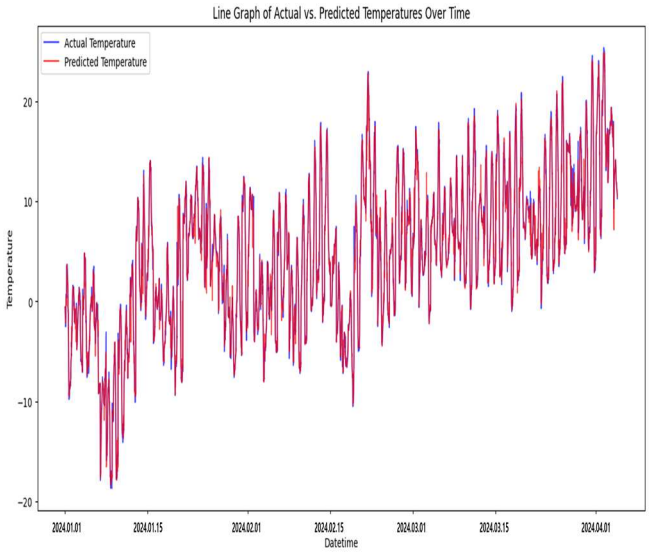

Figure 12. Graph of comparison between actual and predicted values of paprika data using random forest

In the graph above, blue represents the actual values, and red represents the predicted values. However, due to the large amount of data, it is difficult to visually assess how well the actual and predicted values align. Therefore, the prediction accuracy was evaluated using the OOB score. Figure 14 displays the prediction accuracy for each graph.

```
oob_score = oob_model.oob_score_
oob_score
```

```
0.9924099473297654
```

Figure 13. Random Forest Model Paprika Environmental Data OOB Score

The following figure presents the Best RMSE and Test RMSE values obtained through grid search, a hyperparameter tuning technique. Best RMSE refers to the lowest RMSE value achieved during the cross-validation process within the grid search. Test RMSE is calculated by evaluating the final model on a separate test set. RMSE (Root Mean Square Error) is a metric that measures the average error magnitude between predicted and actual values. A lower RMSE indicates a smaller average difference between the predicted and actual values.

```
print(f'Best rmse: {np.sqrt(-grid_search_rf.best_score_):.5f}')
print(f'Test rmse: {m_rmse(grid_search_rf, test_X_imp, test_y):.5f}')
```

```
Best rmse: 0.05668
Test rmse: 0.09316
```

Figure 14. Paprika Environmental Data rmse with Grid Search

Additionally, optimal hyperparameter values were determined using Bayesian optimization, a hyperparameter tuning technique. Figure 15 illustrates the random forest model trained with these optimized values and depicts the differences between the actual and predicted values as evaluated using this model

```
OrderedDict([('bootstrap', True),
             ('max_depth', 26),
             ('max_features', 5),
             ('min_samples_leaf', 1),
             ('min_samples_split', 2),
             ('n_estimators', 150)])
```

```
from sklearn.metrics import mean_squared_error
import numpy as np

model = RandomForestRegressor(**best_params_bayes)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print("Test RMSE:", rmse)
```

```
Test RMSE: 0.7181626856910402
```

Figure 15. Paprika environmental data rmse with Bayesian optimization

Figure 16 presents the analysis of actual and predicted values using a support vector machine, another machine learning algorithm. Based on the RMSE results, it is evident that the predicted values obtained through hyperparameter tuning of the Random Forest model demonstrate higher accuracy compared to those of the support vector machine.

```
(array([10.84251807,  8.18150074,  7.73773995, ..., 12.88238664,
        14.61594436, 11.17594174]),
 6.719053736726161)
```

Figure 16. Paprika environmental data rmse via support vector machine

Figure 17 illustrates the analysis of RMSE between actual and predicted values using the Gradient Boosting model. While the Gradient Boosting model achieves a better RMSE value compared to the support vector machine, the accuracy of the predicted values obtained through hyperparameter tuning of the Random Forest model is higher.

```
(array([17.9530031 ,  6.32624057, 14.55009255, ..., 11.83496511,
        18.27591492, 21.90988513]),
 2.8125357511339932)
```

Figure 17 Paprika environmental data rmse with gradian boosting model

In this study, performance evaluation was conducted using OOB scores obtained with the Random Forest model, along with hyperparameter tuning techniques (grid search and Bayesian optimization), support vector machines, and Gradient Boosting.

Performance evaluation was conducted using the OOB (Out-Of-Bag) score, which is calculated from data not included in the bootstrap sample of each tree. The OOB score provides an estimate of model performance without requiring a separate validation set, making it particularly useful when data is limited. It serves as an alternative to cross-validation; while cross-validation splits the dataset into multiple training and test sets, the OOB score leverages bootstrap samples and their leftover data, offering a more efficient approach.

The OOB scores of the models derived from the previously conducted random forest analysis were 0.9924, demonstrating exceptionally high prediction accuracy for paprika.

RMSE (Root Mean Square Error) is a standard metric used in regression problems to measure the difference between predicted and actual values.

1. Grid Search: Grid search is a hyperparameter tuning method that systematically evaluates a predefined set of

hyperparameter values. By defining a grid of possible hyperparameter combinations, it uses cross-validation to assess model performance for each combination. This exhaustive search ensures highly accurate prediction performance by testing all possible configurations. In model performance evaluation, grid search achieved the best RMSE value of 0.05668, highlighting its effectiveness.

2. Bayesian Optimization: Bayesian optimization is a stochastic model-based optimization technique that is more efficient than grid search, particularly in high-dimensional spaces. It builds a probabilistic model of the function based on the validation set's hyperparameter values and applies criteria such as expected improvement to select new hyperparameter values for evaluation. In the model performance evaluation, Bayesian optimization achieved the second-best RMSE value of 0.71816.

These results demonstrate the effectiveness of both methods, with grid search excelling in achieving optimal performance and Bayesian optimization offering a more efficient alternative for complex spaces.

TABLE II. COMPARED TO HYPERPARAMETER TUNING RMSE VALUES

Classification	***Paprika***
Grid Search	0.05668
Bayesian optimization	0.71816

For the machine learning algorithms, Support Vector Machine and Gradient Boosting, performance was evaluated using RMSE values. The Support Vector Machine achieved an RMSE value of 6.71905, while the Gradient Boosting model produced an RMSE value of 2.81253. Although the RMSE values for both Support Vector Machine and Gradient Boosting are reasonable, the hyperparameter-tuned Random Forest model demonstrated superior performance with a significantly lower RMSE value. This indicates that the model with hyperparameter tuning delivers the best overall performance.

TABLE III. COMPARE MACHINE LEARNING ALGORITHM RMSE VALUES

Classification	***Paprika***
Support Vector Machine	6.71905
Gradient Boosting	2.81253

IV. CONCLUSION

This study focused on developing an artificial intelligence-based horticultural paprika crop growth prediction. Crop growth environment data were collected, and their characteristics were analyzed. Outliers and missing data were identified after visualizing the entire dataset using graphs. Considering the time-series nature of the data, variations were examined across hours, days of the week, and months. The dataset was divided by crop and further split into training, validation, and test sets in an 8:1:1 ratio. Since machine learning and deep learning algorithms often assume similar variances centered around zero, the crop environmental data were scaled before applying the algorithms. The Random Forest model was employed to visualize and compare actual and predicted values for each crop. To improve prediction accuracy, hyperparameter tuning techniques such as grid search and Bayesian optimization were utilized, and RMSE values were calculated. Additionally, RMSE values were obtained using Support Vector Machine and Gradient Boosting algorithms for comparison. In conclusion, the Random Forest model achieved an accuracy of over 90%, demonstrating its ability to provide accurate predictions for real-world farming scenarios. This suggests that systems based on this model can effectively support farmers by offering reliable predictions grounded in real data, thereby optimizing crop growth environments. This research underscores the potential to enhance yields and improve resource utilization, such as water, fertilizers, and pesticides. Future research should extend beyond environmental factors to include predictions of disease outbreaks and pest infestations using historical data. Such advancements could not only improve yields but also address challenges in the agricultural and livestock industries, including the aging workforce and the shortage of young successors in farming. This would contribute to increasing agricultural productivity and sustainability.

ACKNOWLEDGMENT

This work was supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2024-RS-2020-II201489)

REFERENCES

[1] Seo, Y.J., Moon, B.E., & Choi, Y.U. (2022). Analysis of the Structure and Interaction of Smart Farms in Protected Horticulture. Journal of Smart Agriculture, 31(2), 15–22. doi:10.12791/JSA.2022.31.2.15..

[2] Kim, J.H., Lee, S.M., & Park, H.J. (2021). Current Status and Key Challenges of Smart Agriculture in Korea. Korean Journal of Agricultural Economics, 29(3), 45–60. doi:10.12791/KJAE.2021.29.3.45.

[3] Choi, S.Y., & Kim, D.H. (2016). Research Trends and Analysis of ICT Core Technologies for Implementing Smart Farms. Journal of Agricultural Informatics, 25(4), 33–40. doi:10.12791/JAI.2016.25.4.33..

[4] Lee, M.J., & Park, S.H. (2017). A Model Study for Developing Evaluation Criteria for Smart Farms in Protected Horticulture. Journal of Smart Farm Research, 26(1), 11–18. doi:10.12791/JSFR.2017.26.1.11.

[5] Kim, H.S., & Lee, J.Y. (2022). A Study on the Management Performance and Determinants of Smart Farms in Protected Horticulture. Journal of Agricultural Management, 30(2), 25–32. doi:10.12791/JAM.2022.30.2.25.

[6] Park, J.H., & Lee, S.K. (2019). Current Status and Future of Smart Farms in Korea. Korean Journal of Smart Agriculture, 27(3), 55–62. doi:10.12791/KJSA.2019.27.3.55.

[7] Kim, Y.J., & Choi, H.S. (2019). A Study on the Use of Intelligent Smart Farms and Productivity: Focusing on Tomato and Strawberry Hydroponics. Journal of Smart Farming, 28(1), 19–26. doi:10.12791/JSF.2019.28.1.19.

[8] Lee, S.H., & Kim, M.J. (2016). Analysis of the Current Status and Success Factors of Smart Farms. Journal of Agricultural Policy, 24(2), 39–46. doi:10.12791/JAP.2016.24.2.39.

[9] Choi, J.H., & Park, Y.S. (2020). Real-time Temperature Prediction in Cultivation Zones of Protected Horticulture Smart Farms Based on Simple Modeling. Journal of Smart Farm Engineering, 29(3), 27–34. doi:10.12791/JSFE.2020.29.3.27.

[10] Kim, D.H., & Lee, J.S. (2016). Policy and Technology Trends of Korean Smart Farms. Journal of Smart Agriculture Policy, 25(1), 13–20. doi:10.12791/JSAP.2016.25.1.13.