Received: June 1, 2023. Revised: June 29, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Smart Aeroponic Farming System: Using IoT with LCGM-Boost Regression Model for Monitoring and Predicting Lettuce Crop Yield

Gowtham Rajendiran[1]* **Jebakumar Rethnaraj[1]**

[1]*Department of Computing Technologies, School of Computing,*
College of Engineering and Technology, SRM Institute of Science and Technology,
Kattankulathur - 603 203, Chengalpattu, Tamil Nadu, India
* Corresponding author's Email: gr6047@srmist.edu.in

Abstract: Aeroponics is a popular soilless crop cultivation technology that integrates plant nutrition, physiology, and ecological control. It offers automated monitoring, protected cultivation, improved growth mechanisms, better yield and requires less maintenance. Here, to predict the crop yield, two systems are available: manual and automated. Manual systems often fail to produce better prediction results, leading to substantial crop losses whereas, the automated systems use machine intelligence for growth monitoring. This article proposes a lettuce crop growth monitoring-boost (LCGM-Boost) regression model for lettuce yield forecasting in aeroponic vertical farming system. This model is highly robust to outliers, produces better prediction results of 95.86% and lower error rates of 0.36 (MAE), 0.40 (MSE), and 0.63 (RMSE) than other machine learning models namely, support vector, random forest and XGBoost regressors. Hence, it is preferable for growth monitoring and yield prediction of the lettuce crop in the real-time aeroponics system.

Keywords: Aeroponics, Gradient boosting, Growth monitoring, Lettuce, Machine learning regression, Yield prediction.

1. Introduction

Farmland, soil water and labor are the primary factors of the conventional agricultural system [1]. National and international authorities are always looking for innovative ways to boost agricultural productivity in the face of global warming and water shortage. In addition to the advantages of enhancing the production of multiple crops, new technologies have evolved for conserving and utilizing energy [2, 3]. Growing plants with aeroponics entails suspending a container above a grow bed and then spraying the roots with a nutritional solution while keeping them contained in a sealed chamber [4]. Since the nutrient solution is constantly recycled in the growing aeroponic room, monitoring and regularly modifying the pH and EC levels is essential to ensure optimal plant development. Onions, cucumbers, carrots, tomatoes, potatoes, and the lettuce crop all thrive in an EC range of 1.5 to 2.5 ds m-1 and a pH range of 5.5 to 6.5 [5, 6].

High temperatures, excessive light, humidity, and turbidity changes cause plant water evaporation and nutrient loss. These factors can result in a reduction or increase in pH and EC levels [7]. The adjustments to the pH and PPM must be carefully monitored in conventional soilless growing systems like the hydroponic system [8]. Normal pH, EC, and water level modifications are crucial for maintaining the nutrient solution's effectiveness and longevity [9]. Aeroponics is used in places where the soil is unfavorable for plant development, and it has shown a considerable increase in root length, area, volume, and network perimeter [10, 11]. It has been noted that ideal growing conditions of the Aeroponics system include a temperature range of 8 to 44 degrees Celsius and relative humidity of 10 to 94%, which results in an increase in the leaf and root growth of 57%, 42%, and 400% in comparison to the conventional farming method [12].

In contrast, a significant fraction of the world's population now has access to the Internet, which

allows for the widespread use and advancement of IoT technologies to optimize resource utilization in crop production [13]. Considering many problems in aeroponic vertical farming, the two primary hurdles to be focussed on are adjusting the atomization and spraying times of the nutrient solution for each plant sample in the growth tower. Determining the maximum stress level to which each plant may be exposed under the circumstances of irrigation deficit [6, 14] is vital to prevent a drastic drop in root growth and crop productivity. The success of a plant's growth in aeroponics systems depends heavily on the root development process [15, 16].

In consideration of all these factors to be solved by the machine learning algorithms, an improved and expanded version of the gradient boosting machine learning method [31] was developed by T. Chen and C. Guestrin [32] in 2016. This approach is known as extreme gradient boosting-XGBoost. This reduces the likelihood of the model overfitting during training and positively affects the training's convergence speed. More than that, the XGBoost method requires less time to fine-tune the hyper-parameters. XGBoost's rapid computational speed and high accuracy have made it a popular choice for various applications, including data mining and recommendation systems. It is a novel machine-learning technique that may be used to accurately anticipate crop yields in vertical farming.

The research in this work aims to examine and incorporate the efficiency and efficacy of the XGBoost machine-learning algorithm in automating the monitoring process of the aeroponics system. The information gathered by the IoT sensors are analysed in the proposed automated lettuce crop growth monitoring system (ALCGMS). Since lettuce takes a shorter growth time than other crops, it is widely used as a test crop in agricultural experiments. The LCGM is a remote monitoring and management system that enables users to adjust parameters such as atomization duration, visual inspection hours, sprinkler ON/OFF, recirculation system ON/OFF, and nutrient solution mixing. Benefits addressed by the proposed LCGM-Boost regression model include remote monitoring of sensors and actuators built into the agricultural environment, as well as remote capturing of photographs of the crops.

This article is organized so that section 2 presents a brief literature review of the different machine learning algorithms used in the aeroponics system. The configuration and description of the proposed yield prediction system are shown in section 3. The model training, accuracy results and discussions will be presented in section 4 and concluded with the future work described in section 5.

2. Survey of literature

Many related works have been carried out in monitoring the growth of greenhouse plants using IoT systems and machine learning algorithms.

2.1 IoT-oriented agriculture

Kamienski et al. proposed SWAMP, a general architecture for intelligent irrigation management, which combines multiple connecting schemes to disseminate information, implement irrigation distribution models, use drones for visual inspection, use data analysis models, databases, and ensure security for data acquisition [17].

Kaur et al. present a four-layer IoT architecture with sensors and actuators to monitor greenhouses, optimize resource utilization, detect illnesses, identify crop species, optimize irrigation facilities, and utilize pesticides and fertilizers [18].

An alternative five-layer architecture is proposed by Boursians et al., which consists of the physical layer, the datalink layer, the network layer, the authentication layer, and the application layer. The critical value it adds is the capacity to apply machine learning in data analytics and a solar charging system for RF communication devices in the field. Still, it also integrates a weather prediction service to help establish better production methods [19].

The three-layer architecture proposed by Roy et al. consists of sensors and actuators, a remote processing and service layer, and an application layer. Two irrigation strategies are categorized by the stage of the crop's life cycle, and the system is designed to be efficient and user-friendly [20].

2.2 Irrigation systems

Aeroponic crops rely on greenhouse temperature and humidity sensors to regulate watering systems. Lucero et al. set up a three-stage watering schedule based on output days, and their analysis showed that aeroponic systems produced higher yields, leaf area, and root lengths than soil-grown plants [12].

In their presentation of the root chamber's climate control, Jamhari et al. [21] describe the use of a Peltier cell to cool the nutrient solution chamber, keeping the temperature there between 25 and 29oC, and of an ultrasonic humidifier and a fan to keep the relative humidity there between 50% and 70%. Gour et al. [22] suggest using a central processor with an interface between sensors and actuators as well as machine learning capabilities to automate the farming approach.

Belista et al. describe a vertical culture chamber divided into controller and agent modules for crop

care, with the controller responsible for nutritional solution containers, cooling system, evaporative fan, and power supply. Data is stored locally and can be accessed using a mobile app [23].

2.3 Machine learning algorithms

Since its implementation is straightforward and its predictive metrics are easily quantified, the random forest has become a popular algorithm. The system relies on a massive network of interlinked decision trees. By generating and combining many trees, the random forest increases the likelihood of obtaining valid conclusions [24]. The accuracy of the random forest technique is improved by combining the results of several individual tree evaluations. The random forest is used with the GBM to improve the software's accuracy and recycle the analytical method [25].

GBMs are a predictive learning approach that fine-tunes the loss function, identifies weaker learners, and builds a flexible model to improve output and prevent performance problems [26]. The XGB method is a recent development of the regression tree-based gradient boosting machine. The method is based on the idea of "boosting," which entails combining the predictions of multiple "weak" learners into a single "strong" one through additive training procedures. The primary goal of XGB is to lessen the effects of overfitting and under-fitting while decreasing the cost of computation [27].

Linear regression is the most widely used method for making predictions about the correlation between variables. There are two main categories of linear regression: simple and multiple. The dependent variable, x, is always continuous, while the independent variable, y, may be continuous or categorical. Probability distributions and multivariate analysis are used to learn more [28].

SVM is a common supervised learning approach for classifying data and detecting outliers. Models may be constructed in R using the program e1071, which uses a training dataset to predict the classification of an extra data point using a hyperplane that maximizes the spacing between data points in each category (the default is a line). The speed and lack of risk of over-fitting the data make SVM useful [29, 30].

This paper discusses machine learning (ML) algorithms for forecasting lettuce crop production in hydroponic systems with various magnetic water types. Based on input plant and water characteristics, 70% of the datasets were separated for training four ML models (RF, XGB, SVR, and DNNs). For all model situations tested, the R2 was more than 0.77. XGB with scenario 3 has the lowest RMSE, followed by SVR with scenario 3 and RF with scenario 1. SVR with scenario 3 and DNN with scenario 2 were the two best models, however the latter is favoured because to less input variables. By merging input factors with climate variables, the algorithms can be enhanced. For successful crop production prediction on a wide scale, the approaches can be enhanced by integrating input factors with climatic variables, agricultural management data, and better resolution spatiotemporal input variables. ML models might be a quick tool for predicting agricultural production and catastrophe evaluation over a vast area [34].

So, overall from all the literature studies, it is clear that the existing models does not produce the maximum prediction results with respect to the provided lettuce growth dataset. In order to overcome these challenges specifically related to the lettuce yield prediction grown in Aeroponic tower farming, a better Machine Learning model need to be developed without compromising the prediction outputs of the previously designed models with respect to the dataset to address this issue.

3. LCGM-boost regression model

LCGM stands for lettuce crop growth monitoring which follows the characteristics of boosting algorithm. One common kind of boosting method is called "gradient boosting". In this scenario, the error made by the previous predictor is corrected. In contrast to Adaboost, each predictor is trained using the residual errors of the previous one rather than training weights. Gradient boosted tree is a method that uses the CART learner as its foundation (classification and regression trees).

An implementation of the gradient boosted decision trees method in XGBoost is described. To construct decision trees, this approach uses a sequential procedure. The XGBoost algorithm largely relies on weights. Weights are assigned to each independent variable, and this information is then utilized to feed the decision tree, which produces predictions. A decision tree's accuracy is measured by the relative importance of its incorrect predictions, which are then used to inform the tree's subsequent weighting of other factors. The variables whose outcomes were incorrectly predicted by the tree have their weights boosted and are used as inputs to a second decision tree. Combining these several classifiers/predictors produces a more reliable and precise model. It can do standard statistical analyses like regression and classification and more complex ones such as ranking and personalized prediction.

Table 1. Notation list of equation variables

Symbols	Description
$\hat{y}\imath$	predicted value
f_k	Functional space
x_i	Variables (categorical or continuous)
F	Classification and regression trees
$w_q\ (x)$	leaf weight related to sample scoring
$l\left(y_{i,}\ \hat{y}\imath\right)$	Loss function
$\sum_{k=1}^{K}\Omega(f_k)$	Regularization parameter
$\Omega(f_k)$	Complexity of the tree
γ	penalty item of the L1 regularity
λ	penalty item of the L2 regularity
T	number of terminal leaves
ω_j	score in each leaf
$Obj^{(t)}$	Objective function
$f_t\ (x_i)$	Model's prediction for the input data point x_i at iteration t
g_i	Coefficient term associated with $f_t\ (x_i)$
h_i	Another Coefficient term associated with $f_t\ (x_i)$
Gj	Value obtained by summing the variable g over the indices i
Hj	Value obtained by summing the variable h over the indices i
G_L	Gini impurity of the left child after a split
G_R	Gini impurity of the right child after a split
H_L	Number of samples in the left child node after a split
H_R	Number of samples in the right child node after a split

3.1 Mathematics behind the LCGM-boost regression algorithm

Before entering into the mathematics of gradient boosting, here is the notation list of equation variables represented in Table 1 for easy understanding.

Since, the proposed model follows the characteristic features of extreme gradient boosting model, the mathematical derivations of LCGM-boost regression model are as follows:

$$\hat{y}\imath = \sum_{k=1}^{K} f_k\ (x_i),\ f_k \in F \tag{1}$$

where,

k – number of trees, f - functional space of F, F – set of possible classification AND regression trees which can work on both classification (categorical variables) and regression process (continuous variables particularly on time-series data), $\hat{y}\imath$ – the predicted value of the model

Here, the function $f_k\ (x_i)$ can be represented as.

$$f_k\ (x_i) = w_q\ (x) \tag{2}$$

where,

$w_q\ (x)$ is the sample (x) scoring, q is the structure of each tree, w_q is the leaf weight.

Hence, for the model mentioned above, the objective function is provided by,

$$Obj(\theta) = \sum_i^n l\left(y_{i,}\ \hat{y}\imath\right) + \sum_{k=1}^{K}\Omega(f_k) \tag{3}$$

where, $\sum_i^n l\left(y_{i,}\ \hat{y}\imath\right)$ – loss function, which is used for predicting probabilities for binary classification as well as multi-class problems and they are represented as

"binary: logistic,"
"multiclass: softprob"

$\sum_{k=1}^{K}\Omega(f_k)$ – regularization parameter. The constant term acts as the weights, specifically the lambda-L2 regularization. During the gain and weight (prediction) computations, the Hessian is added to the loss function's second derivative. The parameter can also be the 'gamma' value, where the larger the gamma value is, the more conservative the Algorithm will be,

$\Omega(f_k)$ – complexity of the tree, which is represented using the equation number 4,

In this case, the lower the function value, the better the tree's generalization ability and the complexity of the tree is represented as,

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T}\ \omega_j{}^2 \tag{4}$$

where,

γ – penalty item of the L1 regularity, λ - penalty item of the L2 regularity, which is a custom parameter of the Algorithm, T is the number of terminal leaves and ω_j is the score in each leaf.

Here, instead of learning all the trees at once, which complicates optimization, we may use the additive technique, minimize the known loss, and add a new tree, as seen below:

$\widehat{y\imath^0}$ =0
$\widehat{y\imath^1} = f_1\ (x_i) = \widehat{y\imath^0} + f_1\ (x_i)$
$\widehat{y\imath^2} = f_1\ (x_i) + f_2\ (x_i) = \widehat{y\imath^1} + f_2\ (x_i)$
.
.
.
$\widehat{y\imath^t} = \sum_{k=1}^{K} f_k\ (x_i) = \widehat{y\imath^{(t-1)}} + f_t\ (x_i)$

Thus, the objective function of the above model is defined as,

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_i)$$
$$= \sum_{i=1}^{n} l\left(y_i, \hat{y}i^{(t-1)}\right) + f_t(x_i) + \Omega(f_t) + constant$$
$$Obj^{(t)} = \sum_{i=1}^{n} \left(y_i - (\hat{y}i^{(t-1)}) + f_t(x_i)\right)^2 + \sum_{i=1}^{t} \Omega(f_i)$$
$$= \sum_{i=1}^{n} [2\left((\hat{y}i^{(t-1)} - y_i\right) f_t(x_i)) + f_t(x_i)^2] + \sum_{i=1}^{t} \Omega(f_i) + constant$$

Now, let's apply the Taylor series expansion up to the second order; we get:

$$Obj^{(t)} = \sum_{i=1}^{n} [l\left(y_i, \hat{y}i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + +constant \quad (5)$$

where,

$$g_i = \partial \hat{y}i^{(t-1)}\, l\left(y_i, \hat{y}i^{(t-1)}\right) \quad (6)$$

$$h_i = \partial^2 \hat{y}i^{(t-1)}\, l\left(y_i, \hat{y}i^{(t-1)}\right) \quad (7)$$

now, by simplifying and applying Eqs. 6 and 7 in Eq. 5 by removing the constants, we get,

$$Obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \quad (8)$$

Now, we define the regularization term, but first, we need to define the model, and the regularization term is represented below,

From Eq. 4, our objective function becomes,

$$Obj^{(t)} \approx \sum_{i=1}^{n} \left[g_i w_q(x_i) + \frac{1}{2} h_i w_q^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$
$$= \sum_{j=1}^{t} \left[\left(\sum_{i\in I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\omega_j^2\right] + \gamma T \quad (9)$$

Now, we simplify the above expression:

$$Obj^{(t)} = \sum_{j=1}^{T} \left[G_j \omega_j + \frac{1}{2}(H_j + \lambda)\omega_j^2\right] + \gamma T \quad (10)$$

where,

$$Gj = \sum_{i\in Ij} gi \quad (11)$$

$$Hj = \sum_{i\in Ij} hi \quad (12)$$

In this equation, ω_j are independent of each other (Gj and Hj), the best ω_j for a given structure *q(x)* and the best objective reduction is:

$$\omega_j^* = -\frac{Gj}{Hj+\lambda} \quad (13)$$

$$Obj^* = -\frac{1}{2}\sum_{j=1}^{T}\left[\frac{Gj^2}{Hj+\lambda}\right] + \gamma T \quad (14)$$

Gamma is a pruning parameter that reduces the decision tree's size by removing the redundant class of instances, i.e., to do the split operation with the least amount of information gain. This parameter is also responsible for improving the prediction accuracy and reducing the overfitting problem.

In this case, dividing the nodes in a decision tree is required. The greedy algorithm enumerates partitioning schemes by repeatedly starting with a leaf and adding branches to the tree. The decision tree's gain value is the difference between the scores before and after splitting. The ideal split is the one with the highest gain value. After splitting, I_L and I_R are expected to construct an instance set of nodes on the left and right. Eq. 15 gives the split gain value assuming $I = I_L \cap I_R$.

The segmentation with the most considerable gain value is the best,

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] \quad (15)$$

Eq. (15) represents a measure of impurity which is commonly used in the decision tree algorithms, where it aims to minimize the impurity or to maximize the purity in the splitting process.

The architecture of the proposed LCGM-boost regression model for the lettuce crop yield prediction is represented in Fig. 1.

There are four steps of the XGBoost algorithm for monitoring the lettuce crop growth in Aeroponic vertical farming. The steps are listed and explained below:

1) Collection of the data
2) Processing raw data
3) Exploratory data analysis (EDA)
4) Feature engineering

3.2 Collection of the data

The essential growth parameters of the lettuce crop, such as pH, EC, temperature, PPM and turbidity, are collected from the lettuce crop growth aeroponic vertical farming tower of about eight days. The amount of data is about 3432 rows and 5 columns as shown a sample dataset in the Fig. 2. Also The

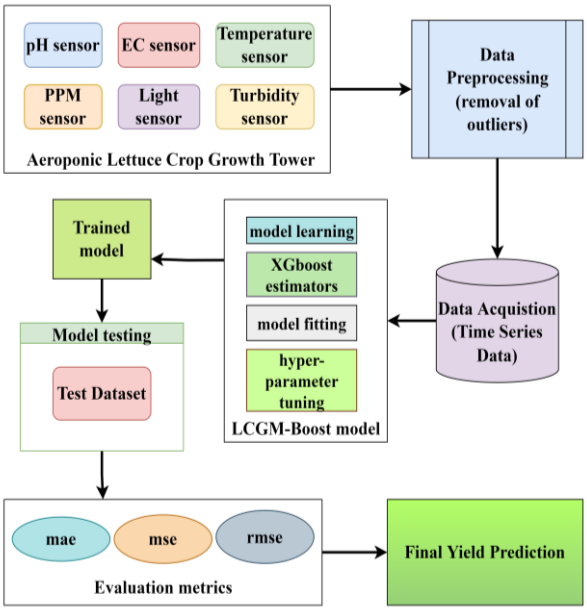


Figure. 1 Lettuce Yield prediction system

	pH	PPM	Temp	EC	Turbidity
0	6	150.0	28.0	0.29	197.0
1	6	953.0	27.0	1.72	196.0
2	6	898.0	27.0	0.28	195.0
3	6	892.0	27.0	1.34	194.0
4	6	819.0	27.0	1.84	193.0

Figure. 2 Sample lettuce crop growing dataset

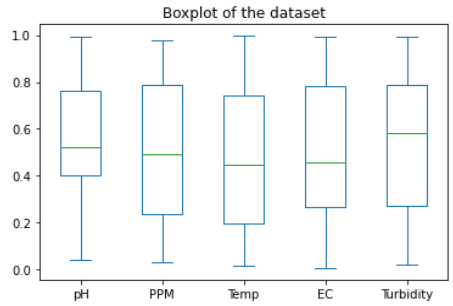
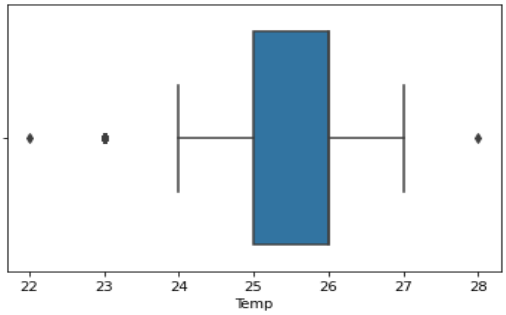


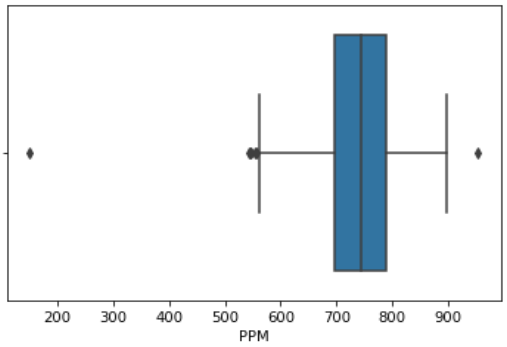
Figure. 3 Dataset visualization using boxplot

utilized time-series dataset is visually represented in the form of a boxplot in Fig. 3 which is used to identify the average value of the data, how dispersed the data is, whether skewness is present in the data, and the presence of outliers in the data.

The above boxplot represents the mean of the distribution of the corresponding dataset. The x-axis represents the parameters such as pH, EC, temperature, turbidity and PPM. In contrast, the y-axis represents the values from 0.0 to 1.0, which denotes the average distribution of the parameters. For example, when considering the parameter pH, the



(a)



(b)

Figure. 4 (a) Outliers representation of temperature and (b) Outliers representation of PPM

average distribution was 0.5, with outliers ranging from 0.1 to 1.0. Similarly, the other parameters were represented.

3.3 Data preprocessing

One of the essential steps in machine learning algorithms is data preprocessing. The time series data must be preprocessed to remove the effect of duplicate, unusual, and missing data from the original data before feeding it into the machine learning model. The steps are as follows:

a) ***Eliminating duplicate data***- The repeated time series data is averaged to reduce the data collection error caused by the sensor. Thus, removing duplicate data will reduce the complexity of data processing by the machine learning model.
b) ***Correct the abnormal data***: Identify the outliers in the data through boxplots. It is observed that the dataset collected may have many abnormal values. So, these abnormal values can be corrected using this preprocessing technique.
c) ***Fill in the missing values***: If the dataset is found to be missing, the necessary values are filled with the appropriate values.
d) ***Fill in the data***-Obtained by cleaning the faults of the test set.
e) ***Removing the outliers***: Though many

preprocessing techniques are available, the method adopted in the research work was eliminating the outliers. The dataset parameters used, namely pH, turbidity, EC, Temperature and PPM concerning the outliers, were visualized in Fig. 4 (a) and (b).

The dataset undergoes the preprocessing technique called removing the outliers to improve the performance of the developed regression model. The prediction results have been briefly discussed in the results and discussion section.

3.4 Exploratory data analysis (EDA)

One of the crucial stages after the data cleaning process is the exploratory data analysis (EDA) which is the graphical representation of the datasets with the help of pairwise plots (other plots are also available) to find the relationships or anomalies to inform the subsequent analysis in the dataset. Though many methods are available in EDA, one of the most overwhelming practical tools is the pairs plot (also called pairwise plot or scatterplot matrix). These pair plots are used to display both the distributions of single variables and the relationship between the two variables.

In the carried research work, the pair plots are implemented using the seaborn data visualization library in the Python language platform Jupyter Notebook. The histogram, which is used to analyze the distribution of a single variable, is shown on the diagonals in the pairs plot. In contrast, the scatter plots on the upper and lower triangles illustrate the relationship between the two variables. The default pairs plot by itself often gives us valuable insights. For the dataset utilized, the pair plots are represented in the next section with the input parameters pH, EC, Temp, Turbidity, and PPM.

3.5 Plots pair grid concerning the correlation coefficient

The mathematical concept known as the correlation coefficient is often used to assess how closely two input variables are related (most preferably in regression techniques). There are three categories of correlation coefficients, namely, positive, negative and no correlation. The positive values indicate a strong positive correlation between the input variables. In contrast, values approximately equal to zero will fall under the category of no correlation and the negative values show a strong negative correlation among the input variables [33].

For illustration, in the utilized dataset, the correlation of the pH data with the other input

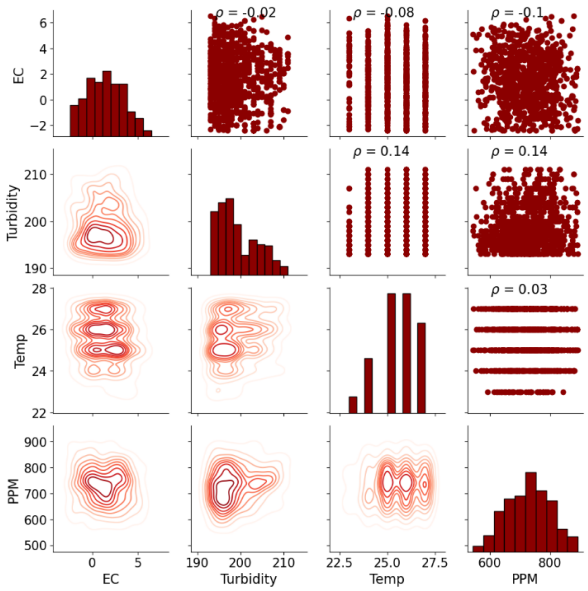


Figure. 5 Visualization of the correlation coefficient of the input parameters

parameters is shown. When the parameter EC in the y-axis is considered, the correlation between the turbidity, temperature, and PPM were -0.02, -0.08 and -0.1, respectively. This indicates no correlation (since zero has no positive or negative sign) with the input parameters since the correlation values were approximately equal to zero. Similarly, when the turbidity and temperature of the y-axis are considered, the correlation values of the temperature and PPM were 0.14, 0.14 and 0.03, respectively, indicating no correlation between the input parameters. Finally, the graphical representation of the correlation coefficient of the scatter plots was displayed above each plot as shown in Fig. 5.

3.6 Feature engineering

Because there are several types of time series data in the indoor vertical farming setting, feature engineering is necessary. To increase prediction reliability and generalization, aspects of preprocessed data must be retrieved, chosen, and organized. First, we extract the polynomial, statistical, aggregate, crossover, and historical information characteristics of pH, EC, temperature, turbidity, PPM, and light. Second and third, the discrete aspects of the data are separated into buckets to improve the model's generalization. The data is organized into buckets based on the hour and day. Then, splice some training set data into the test set to complete the statistical characteristics.

4. Results and discussions

This section deals with the performance of the

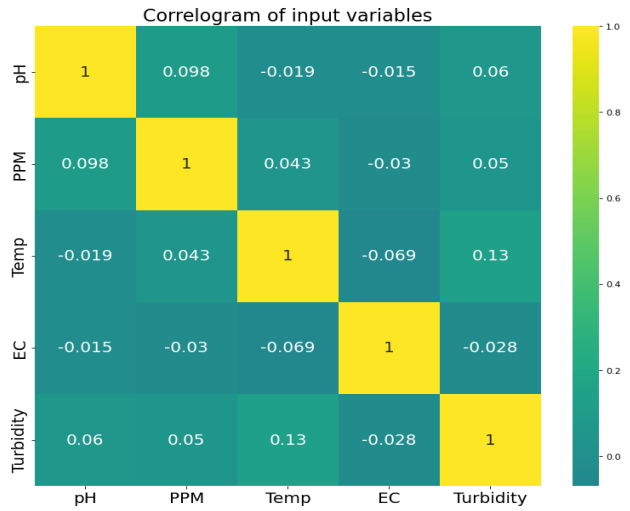


Figure. 6 Dataset visualization using correlogram technique

proposed LCGM-Boost. The section is split into six different sub-sections as follows.

4.1 Platform used

For experimenting with the lettuce crop growth monitoring system, the Anaconda navigator software is used to collect the number of applications, packages, and environments and Python code is used. For the research work, the Jupyter NoteBook platform is mainly used for running the Python code. The XGB regression algorithm is used for the implementation purpose where all the packages needed were available in the list of Python packages. The box was initially imported and the version of the regression algorithm was 1.6.2.

4.2 Data visualization and processing

The next step was exploring the dataset, which included importing, reading and displaying the dataset in the form of graphs. Initially, to visually represent the information about the dataset in graphically (in pictorial form), the package called "matplotlib" was highly utilized. This package was also incorporated to further describe the result analysis at the end of the discussion section. Here, for visualization purpose, correlogram data visualization technique was adopted as shown in Fig. 6.

4.3 Parameter tuning and running the model

Once the dataset was graphically represented, the data splitting was carried out. Here two different variables were used, namely X and y. Both variables were used for training and testing purposes. The first six days of data are used for training, and the next two days are used as the test data set. To optimize the

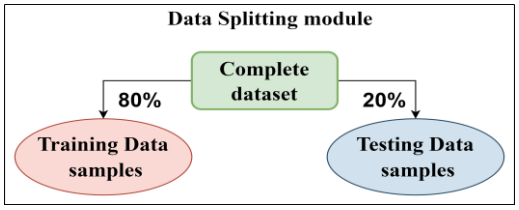


Figure. 7 Dataset splitting

Table 2. Necessary parameters and parametric values

Name of the Parameter	Parameter Value
base_score	0.5
booster	gbtree
colsample_bylevel	1
colsample_bynode	1
colsample_bytree	1
early_stopping_rounds	None
learning_rate	0.300000012
max_cat_to_onehot	4
max_depth	6
max_height	4
min_child_weight	1
n_estimators	100
num_parallel_tree	1
predictor	Auto
random_state	0
reg_lambda	1
verbosity	None

parameters of the XGBoost model, the complete six days of training data are split as follows: four days of data are utilized as the training set, and the following two days are used to validate the model parameters. Fig. 7 depicts the division of the training set, validation set, and test set.

During the dataset splitting, the test size of the dataset was 0.20, i.e., 80% of the dataset was used for training the LCGM-Boost and 20% of the data was used for testing purposes respectively. The reason for 80-20 split-up is that the model showed better prediction results while it showed poor performance for other split-ups. Then, the most critical parameters used by the XGB regressor were displayed, particularly the parameter called n_estimators, whose value was 100.

Next comes the training phase. The XGB model was trained on 80% of the dataset. The important parameters and their respective parametric values, which are used in the regression model, were tabulated in Table 2.

4.4 Grid searchCV

Grid searchCV is a parameter searching technique used in XGB regression to find the best parameter values. It is associated with the cross-

validation approach and tests the model for each combination of values provided in the dictionary. Its main advantage is finding the best solution for hyper-parameter tuning.

4.5 Performance analysis of the LCGM-boost model

The performance metrics used for measuring the performance of the proposed LCGM-boost regression model were MSE, RMSE, MAE. The detail description on the performance metrics were as follows:

a) Mean squared error (MSE): MSE is the average squared difference between the predicted and the actual values of lettuce yield. It is calculated as the average of the squared residuals between the predicted and the actual values, as represented in Eq. (16).

$$MSE = (\frac{1}{n})^{*}\sum(y\_pred - y\_actual)^2 \quad (16)$$

where y_pred is the predicted value of the lettuce yield, y_actual is the actual value of the yield, and n is the number of observations.

b) Root mean squared error (RMSE): As the metric name indicates, it is the square root of the MSE value. It measures the average distance between the predicted and the actual values of the lettuce yield in the same units as the original data. RMSE is a popular metric because it penalizes significant errors more than minor errors and is, therefore, more sensitive to outliers than MSE.

$$RMSE = \sqrt{(MSE)}\ (or)\sqrt{(\frac{1}{n})^{*}\sum(y\_pred - y\_actual)^2} \quad (17)$$

c) Mean absolute error (MAE): MAE measures the average absolute difference between the predicted and actual values of the lettuce yield. It is calculated as the average of the absolute residuals between predicted and actual values.

$$MAE = (\frac{1}{n})^{*}\sum(y\_pred - y\_actual) \quad (18)$$

d) R-squared metric: R^2 ranges from 0 to 1, where a value of 1 indicates the perfect fit and a value of 0 represents no relationship between the variables.

$$R^2 = 1 - [\frac{\sum(y\_actual - y\_pred)^2}{\sum(y\_actual - y\_mean)^2}] \quad (19)$$

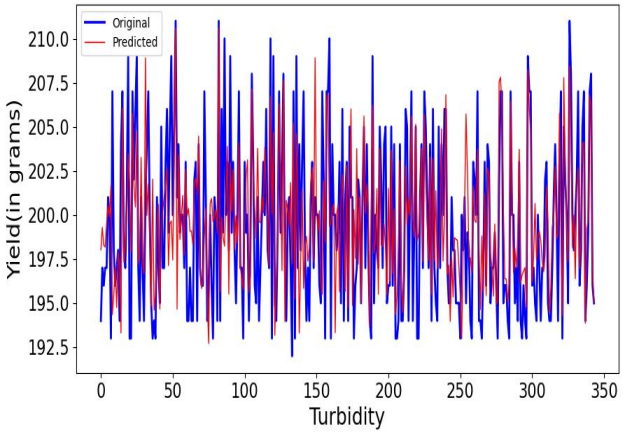


Figure. 8 Prediction graph of LCGM-boost regression model with outliers, without hyper parameter tuning

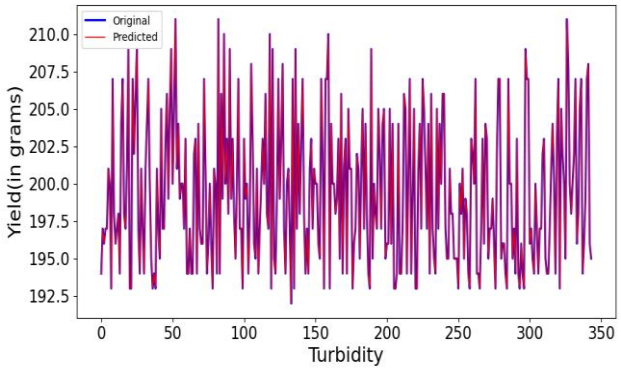


Figure. 9 Final prediction graph of LCGM-boost regression model

where y_pred is the predicted value of the lettuce yield, y_actual is the actual value of the yield, and y_mean is the mean value of the lettuce yield.

4.6 Prediction graphs

The prediction graphs are used to graphically analyse the model's accuracy in yield prediction based on the provided input parameters. A sample of a prediction graph is being represented below for a particular time series dataset "turbidity" and explained in detail.

From Fig. 8, it is seen that the predicted values (denoted in blue color) flow with the actual values (represented in orange color) with slight deviation. So, for the developed regression model based on the dataset provided, the error rates such as MSE, RMSE and MAE were 0.53, 0.72 and 0.96, which is closer to one, indicating that the model has to be tuned further for better performance, i.e., to reduces the scores of those error metrics. This can be done by removing the outliers present in the dataset. Once the preprocessing is done, the outliers will be removed (as discussed in the data preprocessing section).

After removing the outliers, the preprocessed

Table 3. Comparative analysis

Regression Type	MSE	RMSE	MAE	R^2
Support Vector Regressor [34]	1.84	1.36	3.85	0.91
Random Forest [35]	3.49	1.87	1.97	0.92
XGBoost [36]	2.85	1.69	2.2	0.93
LCGM-Boost (Proposed)	0.40	0.63	0.36	0.95

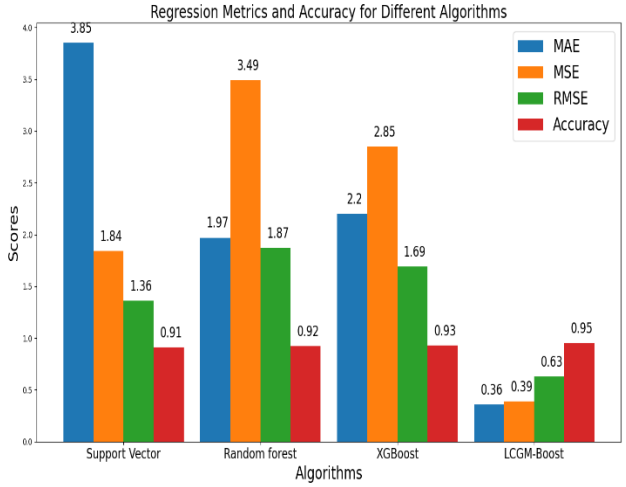


Figure. 10 Comparative results of LCGM-boost regression model with other models

dataset was again fed into the model for predictions. So, after preprocessing, the model produced better prediction results as shown in Fig. 9 with reduced MSE, RMSE and MAE scores of 0.40, 0.63 and 0.36 respectively compared to the dataset before preprocessing. The accuracy of the model was 95.8645% which was comparatively higher. This indicates that the proposed LCGM-boost regression model was better trained on the dataset, validated and tested and produced better prediction accuracy.

4.7 Comparative analysis of LCGM-boost regression model with other regression models

The result outcome of the proposed LCGM-boost regression model were comparatively analyzed with the other regression models such as Support vector regressor [34], Random forest [35], XGBoost [36] respectively as shown in Table 3.

After removing the outliers, the prediction results, such as MSE, RMSE, MAE and R-squared values that are produced by the proposed model are and other regression models were shown in Fig. 10. The proposed work was compared with the other regression models, such as support vector regressor [34] where, the authors used three different scenarios for predicting the yield. The proposed model was compared with random forest model [35], and XGBoost regression model [36].

All these models produced the MSE scores of 1.84, 3.49, 2.85 and 0.40 (Least by LCGM-Boost), RMSE scores of 1.36, 1.87, 1.69 and 0.63 (minimum error rate by LCGM-boost model) while the observed MAE metrics were 3.85, 1.97, 2.2, 0.36 (minimum error rate by LCGM-Boost model) and the R-squared values of 0.91, 0.92, 0.93, 0.95 (maximum value by the LCGM-Boost algorithm) respectively. In all performance metrics, the proposed model outperforms competing models, which are mainly significant values. Hence, this LCGM-boost model can be highly utilized to automate lettuce crop growth monitoring and yield prediction.

5. Conclusion and future scope

The aeroponic crops allow plant roots to be suspended in the air, leading to the lettuce crop growth analysis through the performance results obtained by the LCGM-boost implementation. Here, the growth parameters are not strongly dependent on one another because it is a controlled indoor farming environment. This article proposes a lettuce crop growth monitoring and yield prediction system using the LCGM-boost regression method, which works similarly to the XGBoost algorithm where the considered growth parameters (input) are pH, EC, PPM, Turbidity and temperature. With the help of the proposed model, the growth and yield of an aeroponic lettuce plant can be continuously monitored and predicted by analysing the outcomes of the collected data using the lettuce growth dataset. Also, the LCGM-Boost regression model shows the desired output with better prediction accuracy of 95.86% and the least MSE, RMSE and MAE scores for the chosen lettuce crop. Hence, the proposed regression model is most suitable for automating the lettuce crop growth environment and yield prediction without any doubts.

The present work considered five lettuce growth parameters and provided suggestions for the following researchers. Before transferring to an aeroponic system, plants should go through a normal germination process and hydrogen peroxide should be introduced to the reservoir. The proposed LCGM-boost regression model should be improved by providing other lettuce growth indoor parameters and conducting more numerical experiments. Different transformations are also possible to increase the lettuce crop's productivity within the stipulated period. Those issues would be considered and will be incorporated in the future work. The machine learning models have the potential to be a quick tool for predicting agricultural production and tragedy evaluation across a vast area.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

“Conceptualization, Gowtham Rajendiran and Jebakumar Rethnaraj; methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation Gowtham Rajendiran; writing—review and editing, Gowtham Rajendiran and Jebakumar Rethnaraj; visualization, Gowtham Rajendiran and Jebakumar Rethnaraj; supervision, Jebakumar Rethnaraj; project administration, Jebakumar Rethnaraj”.

Acknowledgements

This work was not supported by any organization and funding agencies.

References

[1] A. J. Hati and R. R. Singh, "Smart Indoor Farms: Leveraging Technological Advancements to Power a Sustainable Agricultural Revolution", *Agri Engineering*, Vol. 3, No. 4, pp. 728-767, 2021.

[2] S. I. Hassan, M. M. Alam, U. Illahi, M. A. A. Ghamdi, S. H. Almotiri, and M. M. Su’ud, "A systematic review on monitoring and advanced control strategies in smart agriculture", *IEEE Access*, Vol. 9, pp. 32517-32548, 2021

[3] Maraveas and T. Bartzanas, "Application of Internet of Things (IoT) for Optimized Greenhouse Environments", *Agri Engineering*, Vol. 3, No. 4, pp. 954-970, 2021.

[4] M. M. Khan, M. T. Akram, R. Janke, R. W. K. Qadri, A. M. A. Sadi, and A. A. Farooque, "Urban horticulture for food secure cities through and beyond COVID-19", *Sustainability*, Vol. 12, No. 22, pp. 9592, 2020.

[5] L. Wimmerova, Z. Keken, O. Solcova, L. Bartos, and M. Spacilova, "A Comparative LCA of Aeroponic, Hydroponic, and Soil Cultivations of Bioactive Substance Producing Plants", *Sustainability*, Vol. 14, No. 4, pp. 2421, 2022.

[6] Lakhiar, J. Gao, T. N. Syed, F. A. Chandio, and N. A. Buttar, "Modern plant cultivation technologies in agriculture under controlled environment: A review on aeroponics", *Journal of Plant Interactions*, Vol. 13, No. 1, pp. 338-352, 2018.

[7] H. Chen, S. Y. Jeng, and C. J. Lin, "Fuzzy logic controller for automating electrical conductivity and pH in hydroponic cultivation", *Applied Sciences*, Vol. 12, No. 1, pp. 405, 2021.

[8] S. Domingues, H. W. Takahashi, C. A. Camara, and S. L. Nixdorf, "Automated system developed to control pH and concentration of nutrient solution evaluated in hydroponic lettuce production", *Computers and Electronics in Agriculture*, Vol. 84, pp. 53-61, 2012.

[9] R. S. V. Gonzalez, A. L. G. Garcia, E. V. Zapata, J. D. O. B. Sanchez, and J. C. S. Savedra, "A Review on Hydroponics and the Technologies Associated for Medium and Small-Scale Operations", *Agriculture*, Vol. 12, No. 5, pp. 646, 2022.

[10] Q. Li, X. Li, B. Tang, and M. Gu, "Growth responses and root characteristics of lettuce grown in aeroponics, hydroponics, and substrate culture", *Horticulturae*, Vol. 4, No. 4, pp. 35, 2018.

[11] Koukounaras, "Advanced greenhouse horticulture: New technologies and cultivation practices", *Horticulturae*, Vol. 7, No. 1, pp. 1, 2020.

[12] L. Lucero, D. Lucero, E. O. Mejia, and G. Collaguazo, "Automated aeroponics vegetable growing system. Case study Lettuce", *IEEE ANDESCON*, pp. 1-6, 2020.

[13] J. C. Negrete, "Internet of things in Mexican agriculture; a technology to increase agricultural productivity and reduce rural poverty", *Research and Analysis Journal*, Vol. 1, No. 2, pp. 1, 2018.

[14] V. Parkash and S. Singh, "A review on potential plant-based water stress indicators for vegetable crops", *Sustainability*, Vol. 12, No. 10, pp. 3945, 2020.

[15] C. B. D. Kuncoro, T. Sutandi, C. Adristi, and Y. D. Kuan, "Aeroponics root chamber temperature conditioning design for smart mini-tuber potato seed cultivation", *Sustainability*, Vol. 13, No. 9, pp. 5140, 2021.

[16] C. M. Nolasco, J. A. P. Medina, J. J. M. Nolasco, R. G. G. Gonzalez, A. I. B. Gutiérrez, and J. J. D. Carmona, "Non-Invasive Monitoring of the Thermal and Morphometric Characteristics of Lettuce Grown in an Aeroponic System through Multispectral Image System", *Applied Sciences*, Vol. 12, No. 13, pp. 6540, 2022.

[17] C. Kamienski, J. P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. S. Cinotti, and A. T. Neto, "Smart water management platform: IoT-based precision irrigation for agriculture", *Sensors*, Vol. 19, No. 2, pp. 276, 2019.

[18] V. P. Kaur and S. Arora, "Recent developments of the Internet of things in agriculture: a survey", *IEEE Access*, Vol. 8, pp. 129924-129957, 2020.

[19] D. Boursianis, M. S. Papadopoulou, A. Gotsis, S. Wan, P. Sarigiannidis, S. Nikolaidis, and S. K.

Goudos, "Smart irrigation system for precision agriculture—The AREThOU5A IoT platform", *IEEE Sensors Journal*, Vol. 21, No. 16, pp. 17539-17547, 2020.

[20] S. K. Roy, S. Misra, N. S. Raghuwanshi, and S. K. Das, "AgriSens: IoT-based dynamic irrigation scheduling system for water management of irrigated crops", *IEEE Internet of Things Journal*, Vol. 8, No. 6, pp. 5023-5030, 2020.

[21] C. A. Jamhari, W. K. Wibowo, A. R. Annisa, and T. M. Roffi, "Design and implementation of IoT system for aeroponic chamber temperature monitoring", In: *Proc. of Third International Conference on Vocational Education and Electrical Engineering*, pp. 1-4, 2020.

[22] M. S. Gour, V. Reddy, M. Vamsi, N. Sridhar, and V. T. Ram, "IoT-based Farming Techniques in Indoor Environment: A Brief Survey", In: *Proc. of Fifth International Conference on Communication and Electronics Systems*, pp. 790-795, 2020.

[23] F. C. L. Belista, M. P. C. Go, L. L. Luceñara, C. J. G. Policarpio, X. J. M. Tan, and R. G. Baldovino, "A smart aeroponic tailored for IoT vertical agriculture using network-connected modular environmental chambers", In: *Proc. of Tenth International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management*, pp. 1-4, 2018.

[24] C. E. Golden, M. J. Rothrock Jr, and A. Mishra, "Comparison between random forest and gradient boosting machine methods for predicting Listeria spp. Prevalence in the environment of pastured poultry farms", *Food Research International*, Vol. 122, pp. 47-55, 2019.

[25] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, "A machine learning-based gradient boosting regression approach for wind power production forecasting: a step towards smart grid environments", *Energies*, Vol. 14, No. 16, p. 5196, 2021.

[26] A. Nagaraju and R. Mohandas, "Multifactor Analysis to Predict Best Crop using Xg-Boost Algorithm", In: *Proc. of Fifth International Conference on Trends in Electronics and Informatics*, pp.155-163, 2021.

[27] G. A. Susto, "A dynamic sampling strategy based on confidence level of virtual metrology predictions", In: *Proc. of Twenty Eighth Annual SEMI Advanced Semiconductor Manufacturing Conference*, pp.78-83, 2017.

[28] S. Venkatesan, V. E. Sathishkumar, J. Park, C. Shin, and Y. Cho, "A Prediction of nutrition water for strawberry production using linear regression", *International Journal of Advanced Smart Convergence*, Vol. 9, No. 1, pp. 132-140, 2020.

[29] S. K. Venkatesan, M. Lee, J. W. Park, C. Shin, and Y. Cho, "A comparative study based on random forest and support vector machine for strawberry production forecasting", *International Journal of Advanced Smart Convergence*, Vol. 9, No. 1, pp. 132-140, 2019.

[30] Y. Ma, S. Zhang, D. Qi, Z. Luo, R. Li, T. Potter, and Y. Zhang, "Driving drowsiness detection with EEG using a modified hierarchical extreme learning machine algorithm with particle swarm optimization: A pilot study", *Electronics*, Vol. 9, No. 5, p. 775, 2020.

[31] J. H. Friedman, "Greedy function approximation: A gradient boosting machine", *Annals of Statistics*, pp. 1189-1232, 2001.

[32] T. Chen and C. Guestrin, "Xgboost: A scalable tree-boosting system", In: *Proc. of 22nd Acm Signed International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016.

[33] Correlation Coefficient: Simple Definition, Formula, Easy Steps. (2021, July). Statistics How To.https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/.

[34] A. Mokhtar, E. S. W. E. Ssawy, H. He, A. A. Nadhir, S. Sh. Sammen, G. A. Yeboah, A. Mohamed, "Using Machine Learning Models to Predict Hydroponically Grown Lettuce Yield", *Frontiers in Plant Science*, pp-13, 2022.

[35] S. M. Basha, D. S. Rajput, J. P. Janet, S. Ramasubbareddy, and S. Ram, "Principles and Practices of Making Agriculture Sustainable: Crop Yield prediction using Random Forest", Vol. 21, No. 4, pp. 591-599, 2020.

[36] G. Idoje, C. Mouroutoglou, T. Dagiuklas, A. Kotsiras, I. Muddesar, and P. Alefragkis, "Comparative analysis of data using machine learning algorithms: A hydroponics system use case", Vol. 4, pp. 100207-100207, 2023.