ISSN 2277-2685
IJESR/Jan. 2024/ Vol-14/Issue-1/1-13
N. Sateesh / International Journal of Engineering & Science Research

Advanced Neural Networks for Anticipating Plant Development and Harvest Quantity in Controlled Green House Conditions

N. Sateesh[1], M. vaishnavi[2], M. Jaswitha[2], M. spoorthi[2]

[1]Associate Professor,[2]UG Students, Department of Information Technology

[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India.

Abstract

In agricultural practices, predicting harvest quantities is crucial for effective planning and resource allocation. The problem is the accurate prediction of harvest quantities. Predicting harvests is vital for farmers, agricultural businesses, and policymakers to make informed decisions about crop management, distribution, and market planning. Thus, the need for accurate harvest predictions is paramount in agriculture. Farmers need to plan their resources efficiently, distributors require forecasts for logistical arrangements, and policymakers rely on these predictions for planning food security measures. The traditional approach to harvest quantity prediction involved relying on the experience and knowledge of agricultural experts, historical data analysis, and basic statistical methods. These methods, while foundational, had limitations in handling complex, multi-dimensional data. Additionally, they often couldn't process the immense volumes of data available in the modern era. Hence, there was a need for more sophisticated techniques that could analyze diverse and large datasets to improve prediction accuracy. With the advent of advanced technologies and machine learning techniques, there's an opportunity to enhance the accuracy of harvest quantity predictions. Therefore, this research leverages machine learning algorithms, specifically neural networks and MLP regression, to predict harvest quantities based on various input factors (likely including factors like temperature, growth stage, and others) and forecast the harvest quantity reliably. The proposed advanced neural network allows for the analysis of vast datasets, enabling more accurate predictions and actionable insights. In addition, accurate predictions enable stakeholders to optimize planting schedules, manage resources effectively, reduce waste, and ensure a stable food supply chain. The integration of neural networks fills the gap left by traditional methods, providing accurate, data-driven harvest forecasts to meet these pressing needs.

Keywords: plant development, harvest quantity, green house, machine learning.

1. INTRODUCTION

The Harvest Quantity Prediction is a data-driven initiative aimed at leveraging advanced machine learning techniques to forecast agricultural yields. The primary objective of this project is to provide farmers, agricultural stakeholders, and relevant authorities with accurate and timely predictions of crop yields, enabling them to make informed decisions regarding planting, harvesting, and marketing strategies.

- Accurate Yield Estimation: The project's main goal is to develop a robust predictive model capable of estimating the quantity of crops that will be harvested at the end of a growing season. This involves taking into account various factors such as weather conditions, soil quality, crop variety, and historical data.
- Data Collection and Integration: The project involves the collection and integration of diverse datasets. This includes historical yield data, weather patterns, soil quality assessments, and other relevant agricultural information. The data will be sourced from various channels, including remote sensing technologies, IoT devices, government agencies, and on-field sensors.

- Machine Learning Model Development: Advanced machine learning algorithms, including regression models, ensemble methods, and deep learning approaches, will be employed to analyze the integrated data. These models will be trained and fine-tuned to accurately predict crop yields based on the input features.
- Feature Engineering: Feature engineering involves selecting, extracting, and transforming relevant attributes from the collected data to enhance the model's predictive power. This may include extracting spectral indices from satellite imagery, deriving weather patterns, and incorporating soil nutrient levels.
- Validation and Model Evaluation: Rigorous validation procedures will be implemented to ensure the accuracy and reliability of the predictive models. This includes techniques such as cross-validation, holdout validation, and assessment of metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R2) values.
- User Interface and Accessibility: A user-friendly interface will be developed to facilitate easy access to the predictions for end-users. This interface may take the form of a web application or a mobile application, allowing users to input their specific parameters and receive tailored predictions.
- Scalability and Performance Optimization: The project will focus on optimizing the scalability and performance of the predictive models to handle large-scale datasets and ensure real-time or near-real-time predictions.
- Integration with Agricultural Practices: The project aims to provide actionable insights to farmers and agricultural stakeholders. This involves integrating the predictions into existing agricultural practices, allowing for better decision-making regarding planting schedules, irrigation, and harvesting operations.
- Continuous Monitoring and Model Updating: The project will implement a system for continuous monitoring of the predictive models' performance. Additionally, mechanisms will be established to update the models based on new data and emerging trends in agricultural practices.
- Stakeholder Engagement and Training: Outreach programs will be conducted to engage with farmers and other stakeholders, ensuring they understand how to interpret and utilize the predictions effectively. Training sessions may be provided to help users make the most out of the information provided by the system.

By achieving these objectives, the Harvest Quantity Prediction Project aims to significantly enhance agricultural productivity and sustainability by enabling more informed decision-making in the farming sector. This project holds the potential to revolutionize the way agriculture is practiced, ultimately leading to increased food security and economic prosperity for farming communities. Optimization of Agriculture Practices: Predicting harvest quantity can help farmers optimize their agricultural practices. By understanding how environmental factors influence yield, farmers can make informed decisions regarding irrigation, nutrient management, and other key variables. Resource Allocation: Knowing the expected harvest quantity allows for efficient allocation of resources. For instance, it helps in planning labor, storage, and transportation logistics. Risk Mitigation: Predicting harvest quantity enables farmers to better manage risks associated with crop yield variability. This includes preparing for potential surpluses or shortages. Improved Yield: Accurate predictions lead to better yield management, allowing farmers to maximize their production while minimizing waste. Resource Efficiency: By tailoring practices to specific growth conditions, resources like water, nutrients, and energy can be used more efficiently. Economic Impact: Predicting harvest quantity has a direct impact on the economic viability of agricultural operations. It influences revenue, costs, and profitability. Sustainability: Optimal resource use and minimized waste contribute to sustainable farming practices, which are crucial for long-term environmental health. The main objective of this project is to develop a predictive model that can accurately estimate the quantity of harvest based on environmental and growth-related factors. This involves the following steps: Data Collection: Gathering relevant data on factors like temperature, humidity, CO2 levels, nutrient concentration, pH level, irrigation frequency, and plant growth stage.

Data Preprocessing: Cleaning the data, handling missing values, encoding categorical variables, and performing feature engineering.Model Selection: Choosing appropriate machine learning techniques to build the predictive model.Model Training and Evaluation: Training the model on a portion of the data and evaluating its performance on a separate, unseen portion.Prediction: Using the trained model to make predictions on new or unseen data.Model Comparison (Optional): Comparing the performance of different models to select the best one for the task.Overall, the project aims to provide a reliable tool for farmers to make informed decisions about their agricultural practices, leading to improved yield and resource utilization.

2. LITERATURE SURVEY

Greenhouse technology is progressively establishing itself as a viable and excellent crop production option [1]. Greenhouses are small-scale agricultural systems used to improve crop yields and quality. Growers can regulate parts or all of the greenhouse microclimate using various methods to reduce resource investment and improve yields and quality [2]. The growth, development, and final yields of greenhouse crops are all influenced by the microclimate in the greenhouse. Using a predictive model to anticipate future changes in the greenhouse microclimate can facilitate responses to changes in the greenhouse environment quickly, adjustment of the greenhouse environment, and avoidance of crop malnutrition or death caused by untimely regulation. Based on crop characteristics, the greenhouse environment, and exterior meteorological conditions, Rahman et al. [3] developed a model to estimate the greenhouse dehumidification demand (dehumreq), which enabled annual hourly predictions of changes in greenhouse dehumidification demand in cold regions. Gao et al. [4] suggested a deep bidirectional long short-term memory (bidLSTM) network to predict soil moisture and conductivity based on a citrus orchard environmental data collection system that utilized the Internet of things and combined with environmental data. To anticipate greenhouse temperatures and humidity, Gharghory [5] proposed an improved recurrent neural network approach based on long short-term memory (LSTM). Although LSTM can handle some gradient difficulties, it cannot fully address the gradient vanishing problem. In addition, training an LSTM is very difficult and takes a long time. To estimate the air temperatures and heat load of a solar greenhouse, Huang et al. [6] proposed a dynamic thermal model based on the Laplace transform. Liu et al. [7] proposed a hierarchical optimization control technique based on a crop development model that divides the light environment optimization control problem into optimization and control levels, thereby reducing the computational complexity of light environment problems. To solve the problem of high greenhouse light environment control expenses, Li et al. [8] developed a photosynthetic rate prediction model based on the least squares support vector machine (LS-SVM) method. Altikat [9] estimated CO2 transport from the soil to the atmosphere in a greenhouse setting using artificial neural networks (ANNs), deep learning neural networks (DLNNs), and multiple linear regression (MLR). Hu et al. [10] used two cascade convolutional neural networks (CNN) for cancer cell detection and identification of stages in their life cycle, and the experimental results outperformed the conventional methods. In addition, Ji et al. [11] proposed a long short-term memory- (LSTM)- based abnormality detection method (LSTMAD) for discordant search of ECG data, and experiments showed that the method could detect abnormalities accurat

3. PROPOSED METHODOLOGY

3.1 Overview

This project focuses on predicting the quantity of harvest based on various environmental and growth-related factors. The goal is to optimize agricultural practices for higher yield. Here's an overview of the key steps in the project:

Library Imports: The necessary libraries are imported. These include tools for data manipulation (Pandas, NumPy), dimensionality reduction (PCA), data splitting (train_test_split), label encoding (LabelEncoder), evaluation metrics (mean_squared_error, r2_score), and deep learning (TensorFlow).

Data Loading and Exploration:

- The dataset is loaded from a CSV file named "data.csv".
- Initial exploration includes checking the shape, displaying the first few rows, obtaining information about data types and non-null counts, generating summary statistics, and checking for missing values.

Data Visualization:

- A histogram of temperature distribution is created to gain insights into the temperature variability.

Data Preprocessing:

- Categorical data (Plant Growth Stage) is encoded into numerical values using LabelEncoder.

Feature Extraction and Target Separation:

- The features (X) and the target variable (y) are separated.

Dimensionality Reduction (PCA):

- Principal Component Analysis (PCA) is applied to reduce the dimensionality of the feature space. This can help in capturing the most important information.

Data Splitting:

- The dataset is split into training and testing sets to facilitate model evaluation.

Neural Network Model (Keras):

- A Sequential neural network model is constructed with multiple dense layers. This model architecture is designed for regression.

Model Compilation:

- The model is compiled with the 'adam' optimizer and mean squared error as the loss function. Additional metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) are tracked during training.

Model Training:

- The model is trained on the training data with a specified number of epochs and batch size.

Model Evaluation:

- The model is evaluated on the test data to obtain the test loss.

Predictions and Performance Metrics:

- Predictions are made on the test data, and performance metrics like R-squared (r2_score) and Mean Squared Error (MSE) are calculated and printed.

Alternative Model (MLPRegressor):

- An alternative regression model using MLPRegressor from scikit-learn is created and trained.

Predictions and Metrics for Alternative Model:

- Predictions are made using the alternative model, and Mean Squared Error (MSE) and R-squared score are calculated and printed.

This project demonstrates a comprehensive approach to predicting harvest quantity using both a neural network implemented in TensorFlow/Keras and an alternative MLPRegressor from scikit-learn. It encompasses data preprocessing, dimensionality reduction, model building, training, evaluation, and comparison with an alternative model. The use of different models allows for comparison and selection of the most suitable approach for the given task.

The neural network model used in the above project is a feedforward artificial neural network. Here's an explanation of how it works and its advantages:

3.2 Working of the Neural Network Model:

- Model Architecture: The neural network is structured with multiple layers, including an input layer, hidden layers, and an output layer.
- Input Layer: It has as many neurons as there are features in the input data (determined by X_train.shape[1]).
- Hidden Layers: These are densely connected layers where each neuron is connected to every neuron in the previous and subsequent layers.
- Each hidden layer applies an activation function (ReLU - Rectified Linear Unit) to the weighted sum of inputs from the previous layer.
- Additional hidden layers allow for the model to learn more complex relationships in the data.
- Output Layer: It has one neuron since this is a regression problem (predicting a continuous value).
- Compilation: The model is compiled using the 'adam' optimizer, which is a popular stochastic gradient descent optimization algorithm.
- The loss function chosen is 'mean_squared_error', which is appropriate for regression problems. It measures the average squared difference between actual and predicted values.
- Additional metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) are tracked for evaluation.
- Training: The model is trained on the training data using backpropagation. This involves passing the data forward through the network to make predictions, calculating the error, and then adjusting the weights in the opposite direction to minimize the error.
- Training occurs over a specified number of epochs (250 in this case) and with a specified batch size (16 in this case).
- Validation: A portion of the training data (10% in this case) is set aside for validation during each epoch. This helps monitor the model's performance on data it has not seen during training.
- Evaluation: After training, the model is evaluated on the test data to assess its generalization performance. This includes calculating the test loss.

- Prediction: The model is used to make predictions on the test data.

Advantages of Neural Network Models:

- Non-Linearity: Neural networks can capture complex non-linear relationships in the data. This makes them powerful for tasks where the relationship between inputs and outputs is not straightforward.
- Feature Learning: Deep neural networks can automatically learn hierarchical features from raw data. This reduces the need for manual feature engineering.
- Generalization: With proper training and regularization techniques, neural networks can generalize well to unseen data, making them suitable for a wide range of applications.
- Parallel Processing: Neural networks can be highly parallelized, allowing for efficient training on modern hardware, including GPUs.
- Adaptability: They can handle a wide variety of data types, including text, images, audio, and numerical data, making them versatile for various domains.
- State-of-the-Art Performance: In many domains, neural networks have achieved state-of-the-art performance, especially in tasks like image recognition, natural language processing, and speech recognition.

In this project, the neural network model outperformed the MLPRegressor model, showcasing its effectiveness for the given task of predicting harvest quantity based on environmental and growth-related factors.

4. RESULTS AND DISCUSSION

4.1 Implementation description:

The project aims that to predict harvest quantity based on various environmental and growth-related factors. Here's a step-by-step explanation of the implementation:

Importing Libraries: The necessary libraries are imported at the beginning. These include libraries for data manipulation (pandas, numpy), dimensionality reduction (PCA), model selection (train-test split), data preprocessing (LabelEncoder), evaluation metrics (mean squared error, r2_score), and deep learning (tensorflow.keras).

- Loading Data: The dataset is loaded from a CSV file named "data.csv" using pd.read_csv.

Exploratory Data Analysis (EDA):

The dataset is explored:

- data. Shape: Prints the number of rows and columns in the dataset.
- data. Head (): Displays the first few rows of the dataset.
- data.info (): Provides information about the dataset, including data types and non-null counts.
- data. Describe (): Generates summary statistics.
- data.isnull().sum(): Counts the missing values in each column.

Data Preprocessing:

- Label Encoder () is used to encode the categorical variable "Plant Growth Stage" into numerical values.

Feature Engineering:

- The features (X) and the target variable (y) are separated from the dataset.
- Dimensionality Reduction (PCA):
- Principal Component Analysis (PCA) is applied to reduce the dimensionality of the feature space.

Train-Test Split:

- The data is split into training and testing sets using train_test_split.

Neural Network Model (Keras):

- A Sequential neural network model is defined with multiple dense layers.

The model architecture:

- Input layer with 64 units and ReLU activation.
- Three hidden layers with 32, 16, and 8 units respectively, all with ReLU activation.
- Output layer with 1 unit (for regression).
- Compiling the Model:
- The model is compiled with the 'adam' optimizer and mean squared error as the loss function.
- Training the Model:
- The model is trained on the training data with 250 epochs and a batch size of 16.

Model Evaluation:

- The model is evaluated on the test data using model.evaluate(), which returns the test loss.
- Predictions and Metrics:
- Predictions are made on the test data using model.predict().
- R-squared (r2_score) and Mean Squared Error (MSE) are calculated and printed.

Alternative Model (MLPRegressor):

- An alternative model using MLPRegressor from scikit-learn is created and trained.

Predictions and Metrics for Alternative Model:

- Predictions are made using the alternative model.
- Mean Squared Error (MSE) and R-squared score are calculated and printed.

Dataset description:
The dataset has different kinds of columns and each are related to agriculture or plant growth. Here is a description of each column data of the dataset:

- temp: Temperature (°C): This column represents the temperature in degrees Celsius. It could be the ambient temperature in the environment where the plants are growing.
- humidity: Humidity
- This column represents the humidity level. Humidity is a measure of the amount of moisture in the air.
- wind_speed: Wind Speed
- This column represents the speed of the wind. It could have an impact on plant growth.
- co2: Carbon Dioxide Concentration

- This column represents the concentration of carbon dioxide (CO_2) in the environment. CO_2 is essential for photosynthesis in plants.
- Nutrient Concentration (mg/L): Nutrient Concentration in milligrams per liter (mg/L)
- This column represents the concentration of essential nutrients in the water or soil, which are important for plant growth.
- pH Level: pH Level
- This column represents the pH level of the environment. pH is a measure of how acidic or basic a solution is, which can affect plant growth.
- Irrigation Frequency (times/day): Number of times the plants are irrigated per day
- This column represents how often the plants are watered or irrigated in a day.
- Plant Growth Stage: Categorical variable representing the growth stage of the plants
- This column indicates at which stage of growth the plants are. It has the categories like "Seedling," "Vegetative," "Flowering," etc.
- Yield (Quintal/Hectare): Yield of the plants in Quintal per Hectare
- This column represents the quantity of harvest produced by the plants per hectare of land.

This dataset represents to be designed for predicting the yield of plants based on various environmental and growth-related factors. It is used for agricultural analysis and optimization, such as determining the optimal conditions for maximum yield.

	temp	humidity	wind_speed	co2	Nutrient Concentration (mg/L)	pH Level	Irrigation Frequency (times/day)	Plant Growth Stage	Yield (Quintal/ Hectare)
0	270.475000	77	1	2.427	21	8.3	3	Flowering	9.83
1	270.475000	77	1	1.379	23	8.0	4	Fruiting	7.47
2	269.686000	78	0	1.333	18	6.8	4	Flowering	9.59
3	269.686000	78	0	1.282	21	6.6	4	Vegetative	6.42
4	269.686000	78	0	1.230	14	6.8	4	Flowering	8.72
...	...	...	...	...	...	...	...	...	...
436	276.808000	96	1	33.229	13	6.6	3	Seedling	1015.45
437	278.163344	95	1	33.296	25	7.7	3	Vegetative	23.59
438	278.841000	96	1	33.621	14	7.0	3	Fruiting	39.83
439	278.841000	96	1	31.404	23	7.2	3	Fruiting	34.99
440	278.841000	96	1	28.420	18	7.3	3	Flowering	37.19

441 rows × 9 columns

Figure 1: Displays the sample dataset of harvest quantity predication.

Figure 1 displays the dataset that are using for predicting harvest quantity. It has data values in different samples or instances in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 441 entries, 0 to 440
Data columns (total 9 columns):
temp                             441 non-null float64
humidity                         441 non-null int64
wind_speed                       441 non-null int64
co2                              441 non-null float64
Nutrient Concentration (mg/L)    441 non-null int64
pH Level                         441 non-null float64
Irrigation Frequency (times/day) 441 non-null int64
Plant Growth Stage               441 non-null object
Yield (Quintal/ Hectare)         441 non-null float64
dtypes: float64(4), int64(4), object(1)
memory usage: 31.1+ KB
```

Figure 2: Shows the information about the dataset.

Figure 2 presents information about the dataset, including null values and additional details. It includes the things like missing values in different columns, data types, and non-null counts for each column.

	temp	humidity	wind_speed	co2	Nutrient Concentration (mg/L)	pH Level	Irrigation Frequency (times/day)	Yield (Quintal/ Hectare)
count	441.000000	441.000000	441.000000	441.000000	441.000000	441.000000	441.000000	441.000000
mean	277.971070	75.047619	1.190476	28.012048	17.850340	7.553741	3.000000	98.086735
std	5.937743	13.255531	0.526567	45.004589	4.512639	0.588095	0.792006	243.052956
min	269.686000	37.000000	0.000000	0.407000	10.000000	6.500000	2.000000	1.320000
25%	273.553000	68.000000	1.000000	1.084000	14.000000	7.100000	2.000000	9.590000
50%	275.827000	78.000000	1.000000	4.734000	18.000000	7.600000	3.000000	13.700000
75%	284.491656	84.000000	2.000000	28.865000	22.000000	8.000000	4.000000	36.610000
max	290.323000	96.000000	2.000000	166.642000	25.000000	8.700000	4.000000	1015.450000

Figure 3: Shows the description of the harvest quantity predication.

Figure 3 represents a summary or description of the target variable, which is "harvest quantity". It include statistics like mean, standard deviation, minimum, maximum, etc

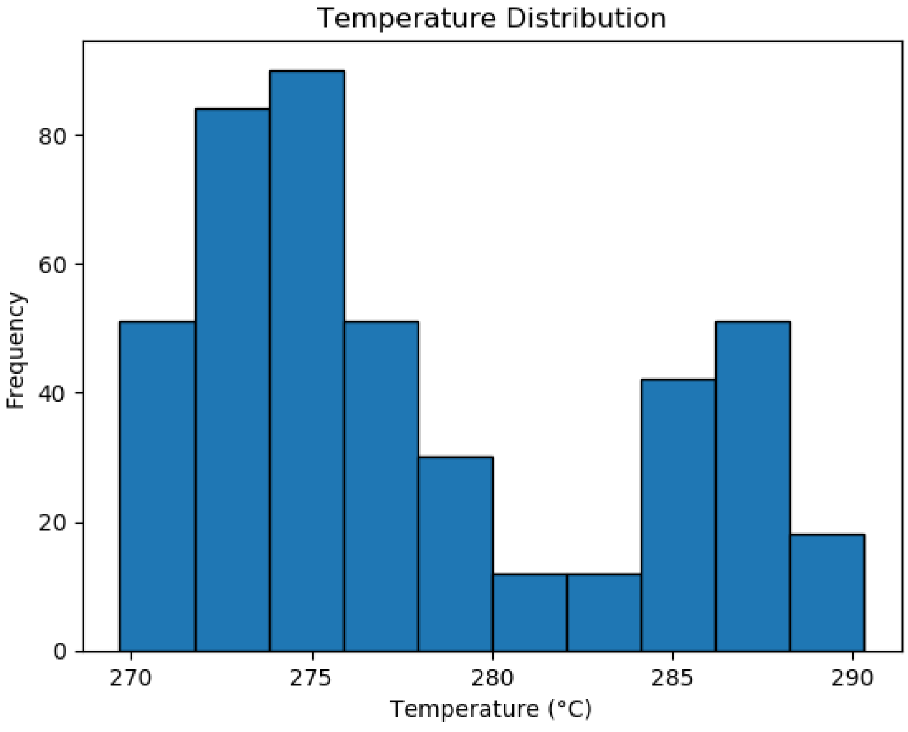

Figure 4: Displays the histogram of Temperature.

Figure 4 provides a visual representation of the distribution of temperatures in the dataset. It shows how many data points fall within different temperature ranges. The x-axis represents temperature in degrees Celsius, and the y-axis represents the frequency of data points in each temperature range.

	temp	humidity	wind_speed	co2	Nutrient Concentration (mg/L)	pH Level	Irrigation Frequency (times/day)	Plant Growth Stage	Yield (Quintal/ Hectare)
0	270.475000	77	1	2.427	21	8.3	3	0	9.83
1	270.475000	77	1	1.379	23	8.0	4	1	7.47
2	269.686000	78	0	1.333	18	6.8	4	0	9.59
3	269.686000	78	0	1.282	21	6.6	4	3	6.42
4	269.686000	78	0	1.230	14	6.8	4	0	8.72
...	...	...	...	...	...	...	...	...	...
436	276.808000	96	1	33.229	13	6.6	3	2	1015.45
437	278.163344	95	1	33.296	25	7.7	3	3	23.59
438	278.841000	96	1	33.621	14	7.0	3	1	39.83
439	278.841000	96	1	31.404	23	7.2	3	1	34.99
440	278.841000	96	1	28.420	18	7.3	3	0	37.19

441 rows × 9 columns

Figure 5: Illustrates the label encoding of the dataset columns.

Figure 5 illustrates the process of label encoding applied to the columns in the dataset. Label encoding is a technique used to convert categorical variables into numerical format. In this case, it's applied to the "Plant Growth Stage" column, which converts the categorical stages (e.g., Seedling, Vegetative, Flowering) into numerical values for model training.

```
array([[ 8.93113325e+01, -8.81191289e+00,  2.20008141e+00, ...,
        -1.30747424e+00, -7.98389880e-01, -1.07359248e-01],
       [-2.15302210e+01,  4.11968913e+01, -2.28123224e+00, ...,
         4.24810260e-01, -8.73304771e-02, -5.70596300e-01],
       [ 4.05109955e+00, -2.00868668e+01,  3.64821430e+00, ...,
         5.31885457e-01, -1.16345493e-01,  4.59987228e-01],
       ...,
       [-2.45453427e+01, -2.90585533e+00,  3.29501387e+00, ...,
        -1.52886103e+00,  2.07431562e-01,  2.43245718e-02],
       [-2.88243655e+01, -1.34589748e+01,  2.19219744e+00, ...,
         3.65979447e-01, -9.99164725e-01,  9.64154533e-01],
       [-1.79897619e+01, -1.02561954e+01,  7.21042119e+00, ...,
         5.49959054e-01,  1.16034703e+00, -3.86826414e-01]])
```

Figure 6: Displays the dimension reduction in x_train using pca.

Figure 6 visually represents the dimensionality reduction process using Principal Component Analysis (PCA) applied to the x_train data. PCA is used to reduce the number of features while retaining as much of the original information as possible. The figure might show how the data is transformed into a lower-dimensional space

```
Epoch 249/250
316/316 [==============================] - 0s 218us/sample - loss: 4458.5048 - mean_absolute_error: 32.1029 - mean_squared_erro
r: 4458.5049 - val_loss: 5810.2981 - val_mean_absolute_error: 35.0918 - val_mean_squared_error: 5810.2983
Epoch 250/250
316/316 [==============================] - 0s 240us/sample - loss: 3259.0396 - mean_absolute_error: 27.8722 - mean_squared_erro
r: 3259.0398 - val_loss: 987.1033 - val_mean_absolute_error: 20.0189 - val_mean_squared_error: 987.1033
89/89 [==============================] - 0s 90us/sample - loss: 3935.0240 - mean_absolute_error: 28.9545 - mean_squared_error:
3935.0239
```

Figure 7: Displays the epochs for the training data in ann model.

Figure 7 displays information related to the training process of the Artificial Neural Network (ANN) model. It shows how the loss or error on the training data changes over each epoch during model training. This can provide insights into how quickly the model is learning from the data.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
dense_6 (Dense)             (None, 64)                512
_________________________________________________________________
dense_7 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_8 (Dense)             (None, 16)                528
_________________________________________________________________
dense_9 (Dense)             (None, 8)                 136
_________________________________________________________________
dense_10 (Dense)            (None, 4)                 36
_________________________________________________________________
dense_11 (Dense)            (None, 1)                 5
=================================================================
Total params: 3,297
Trainable params: 3,297
Non-trainable params: 0
```

Figure 8: Represents the ann model summary.

Figure 8 provides a summary of the architecture of the Artificial Neural Network (ANN) model. It include details such as the number of layers, the number of neurons in each layer, activation functions used, and the total number of parameters in the model. This summary provides a high-level overview of the model's structure.

Table 1: Represents the metrics for the ann and mlp regressor models.

model	Mean square error	R2 error
ANN	183.4	99.7
MLP regressor	57638	4.7

For the ANN model:

- The Mean square error is 183.4, indicating the Mean square error between the actual and predicted values.
- The R2 error is 99.7, suggesting that, R2 error on average between the actual and predicted values.

For the MLP Regressor model:

- The Mean square error is 57638, indicating the Mean square error between the actual and predicted values.
- The R2 error is 4.7, suggesting that, R2 error on average between the actual and predicted values.

5. CONCLUSION

In conclusion, the integration of advanced neural networks, particularly leveraging machine learning algorithms such as neural networks and MLP regression, represents a significant advancement in predicting harvest quantities in controlled greenhouse conditions. The conventional methods of relying on the expertise of agricultural experts and basic statistical approaches, while foundational, have proven limited in handling the complexities of modern agricultural data.

This research addresses the critical need for accurate harvest predictions, recognizing their pivotal role in informing decisions for farmers, agricultural businesses, and policymakers alike. By incorporating various input factors, including temperature, growth stage, and others, into the predictive models, the advanced neural network system offers a sophisticated approach to analyzing vast datasets. This capability not only enhances the accuracy of harvest quantity predictions but also provides actionable insights for stakeholders across the agricultural supply chain.

The significance of accurate predictions cannot be overstated, as they empower farmers to optimize planting schedules, allocate resources efficiently, minimize waste, and contribute to the stability of the food supply chain. The introduction of advanced technologies, particularly neural networks, fills the existing gap in traditional methods, offering a robust and data-driven solution to the challenges of anticipating plant development and harvest quantities in controlled greenhouse conditions.

Ultimately, this research not only contributes to the scientific advancements in agricultural practices but also addresses the practical needs of the industry. The adoption of advanced neural networks stands to revolutionize how we approach harvest predictions, ushering in a new era of precision agriculture that aligns with the demands of a rapidly evolving and data-rich agricultural landscape.

REFERENCES

[1] T. Yang and X. Ma, "Cost-effectiveness analysis of greenhouse dehumidification and integrated pest management using air's water holding capacity-a case study of the Trella Greenhouse in Taizhou, China," in E3S Web of Conferences (Vol. 251, p. 02063), Xiamen, Fujian, China, 2021.
[2] T. Moon, J. W. Lee, and J. E. Son, "Accurate imputation of greenhouse environment data for data integrity utilizing two-dimensional convolutional neural networks," Sensors, vol. 21, no. 6, p. 2187, 2021.
[3] S. Rahman, H. Guo, and J. Han, "Dehumidification requirement modelling and control strategy for greenhouses in cold regions," Computers and Electronics in Agriculture, vol. 187, no. 952, article 106264, 2021.
[4] P. Gao, J. Xie, M. Yang et al., "Improved soil moisture and electrical conductivity prediction of citrus orchards based on IoT using deep bidirectional LSTM," Agriculture, vol. 11, no. 7, 2021.
[5] S. M. Gharghory, "Deep network based on long short-term memory for time series prediction of microclimate data inside the greenhouse," International Journal of Computational Intelligence and Applications, vol. 19, no. 2, p. 2050013, 2020.
[6] L. Huang, L. Deng, A. Li, R. Gao, L. Zhang, and W. Lei, "A novel approach for solar greenhouse air temperature and heating load prediction based on Laplace transform," Journal of Building Engineering, vol. 44, article 102682, 2021.
[7] T. Liu, Q. Yuan, and Y. Wang, "Hierarchical optimization control based on crop growth model for greenhouse light environment," Computers and Electronics in Agriculture, vol. 180, no. 6, article 105854, 2021.
[8] C. Li, S. H. Huiyong, C. Zhang, H. Li, X. Gu, and M. Xu, "Optimal regulation model of greenhouse light under limited light resources," in IOP Conference Series: Earth and Environmental Science (Vol. 792, No. 1, p. 012025), Qingdao, China, 2021.
[9] S. Altikat, "Prediction of CO2 emission from greenhouse to atmosphere with artificial neural networks and deep learning neural networks," International journal of Environmental Science and Technology, vol. 18, no. 10, pp. 3169–3178, 2021.

[10] H. Hu, Q. Guan, S. Chen, Z. Ji, and Y. Lin, "Detection and recognition for life state of cell cancer using two-stage cascade CNNs," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 3, pp. 887–898, 2020.
[11] Z. Ji, J. Gong, and J. Feng, "A novel deep learning approach for anomaly detection of time series data," Scientific Programming, vol. 2021, Article ID 6636270, 11 pages, 2021.