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

Shaik Fiaza Tazeen[1], D.Murali[2]
[1]*PG Student, QUBA College of engineering and technology*
[2]*Associate Professor, QUBA College of engineering and technology*

Abstract: In the realm of controlled greenhouse agriculture, precise anticipation of plant development stages and harvest quantities is indispensable for maximizing productivity. This paper introduces an ADVANCED NEURAL NETWORK (ANN)-based approach for this purpose, contrasting its performance with the existing system utilizing the Random Forest algorithm. Our proposed neural network model harnesses the capabilities of recurrent neural networks to capture temporal and spatial dependencies within greenhouse data. By integrating diverse inputs such as environmental parameters (e.g., temperature, humidity, light intensity), soil conditions, and plant physiological data, the model adeptly learns intricate patterns associated with plant growth stages and yield outcomes. Moreover, attention mechanisms are embedded within the neural network architecture, facilitating dynamic feature selection and enhancing prediction accuracy. Through comprehensive experimentation and validation on authentic greenhouse datasets, we demonstrate the superior performance of our neural network model compared to the incumbent Random Forest(RF)-based system. The findings underscore the efficiency of ANN in accurately forecasting plant development stages and estimating harvest quantities, surpassing the predictive capabilities of RF.

Index Terms— **credit card fraud, online transactions, machine learning, Random Forest, data preprocessing, fraud detection, GridSearchCV, real- time prediction, transaction behavior, model performance, hyperparameter tuning, interactive interface, financial losses, fraud trends, visualizations, accuracy, computational efficiency.**

1. INTRODUCTION

1.1 Overview

In modern agriculture, greenhouse cultivation has emerged as a preferred method due to its ability to extend growing seasons and provide a controlled environment that safeguards Crops against weather fluctuations. However, accurately forecasting crop yields within greenhouses poses a significant challenge due to the complex interplay of factors such as temperature, humidity, carbon dioxide levels, and disease occurrences.

To address this challenge, we propose a neural network-based approach for greenhouse crop yield prediction. Our method leverages the power of two cutting-edge neural network architectures. By integrating these models, we aim to capture the temporal dependencies within the input data sequences, including historical greenhouse parameters and past yield information.

1.2 History

Artificial intelligence (AI) technologies have rapidly evolved since the late 20th century, with machine learning and deep learning algorithms becoming increasingly sophisticated. These advancements have enabled AI systems to process vast amounts of data and perform complex tasks, ranging from image recognition to natural language processing. As AI continues to mature, its applications span across various sectors, including finance, healthcare, and manufacturing.

The deployment of neural networks has revolutionized industries such as image and speech recognition, autonomous vehicles, and recommendation systems. In healthcare, they facilitate disease diagnosis and drug discovery, while in finance, they enable fraud detection and risk assessment.

As neural network technologies continue to advance, researchers and developers are exploring novel architectures and training techniques to further improve performance and efficiency. The future holds promise for even more innovative applications of neural networks, driving progress and innovation across various domains.

1.3 Problem Statement

In modern agriculture, controlled greenhouse environments are prized for optimizing plant growth and yield. However, accurately anticipating plant development and harvest quantity remains challenging, reliant on manual observation prone to errors. Plant growth dynamics are complex, influenced by factors like light, temperature, and nutrients, demanding sophisticated algorithms. Greenhouse environments are variable due to seasonal changes and equipment malfunctions, requiring robust neural networks. Diversity of plant species adds complexity, necessitating models accommodating various crops while maintaining accuracy. Dynamic interactions between plants and their environment further complicate predictions, requiring continuous adaptation. Obtaining labeled training data is limited by resources, requiring alternative collection methods. ANN architectures like RNNs and CNNs capture complex growth patterns.

II SYSTEM ANALYSIS

2.1 LITERATURE SURVEY

Jähne et.al [1] The model strain for Gram-positive plant growth-promoting bacteria was identified. Genome mining revealed that at least 15 natural product biosynthesis gene clusters (BGCs) were well conserved in all B. velezensis strains. In total, 36 different BGCs were identified in the genomes of the strains representing B. velezensis, B. subtilis, Bacillus tequilensis, and Bacillus altitudinis. In vitro and in vivo assays demonstrated the potential of the B. velezensis strains to enhance plant growth and to suppress phytopathogenic fungi and nematodes. Due to the promising potential to stimulate plant growth and to support plant health, the B. velezensis strains TL7 and S1 were selected as starting material for the development of novel biostimulants and biocontrol agents efficient in protecting the important Vietnamese crop plants black pepper and coffee against phytopathogens. The results of the large-scale field trials performed in the Central Highlands in Vietnam corroborated that TL7 and S1 are efficient in stimulating plant growth and protecting plant health in large-scale applications. It was shown that treatment with both bioformulations resulted in prevention of the pathogenic pressure exerted by nematodes, fungi, and oomycetes, and increased harvest yield in coffee and pepper.

Mohmed et.al [2] The paper proposed an Artificial Neural Network (ANN) model to predict plant response to environmental conditions to enhance crop production systems that improve plant performance and resource use efficiency (e.g. light, fertilizer, and water) in a Chinese Solar Greenhouse. Comprehensive data collection had been conducted in a greenhouse environment to validate the proposed prediction model. Specifically, the data had been collected from the CSG in warm and cold weather.

III EXISTING METHODOLOG

3.1 Introduction

This section sets the stage by explaining the purpose and objectives of the project. It addresses the need for advanced predictive models in controlled greenhouse environments to optimize plant development and harvest quantity. It was also include background information on the importance of efficient greenhouse management for agricultural productivity and sustainability.

3.2 Existing System Overview

This part provides an analysis of the current methods and systems used in controlled greenhouse environments for predicting plant development and harvest quantity. It discusses the shortcomings and limitations of traditional approaches, underscoring the necessity for more sophisticated models like neural networks. It details the processes involved in the system, such as data collection from various sensors (temperature, humidity, light intensity), preprocessing steps (cleaning, normalization), model training (using neural network architectures), and evaluation methodologies.

3.3 Random Forest Algorithm

RF is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "RF is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the RF takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

ALGORITHM:

Step-1 Select random K data points from the training set.

Step-2 Build the decision trees associated with the selected data points (Subsets).

Step-3 Choose the number N for decision trees that you want to build.

Step-4 Repeat Step 1 & 2.

Step-5 For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

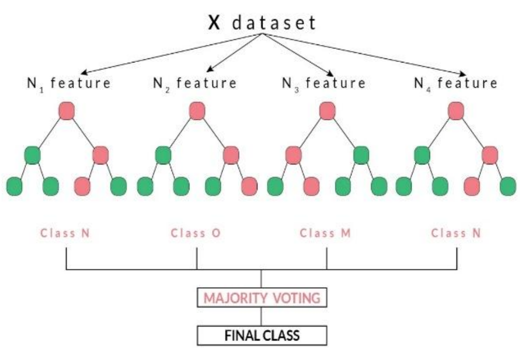


Fig 3.1: Random forest

3.4 Implementation Details

Here are the steps to implement the RF algorithm for predicting plant development and harvest quantity in controlled greenhouse conditions:

Import Libraries: Import the necessary libraries, including pandas for data manipulation, numpy for numerical operations, and scikit-learn for machine learning functionalities.

Load and Prepare Data: Load your dataset using Pandas and perform any necessary preprocessing steps such as handling missing values, encoding categorical variables, and splitting the data into features (X) and target variable (y).

Split Data into Training and Testing Sets: Split your dataset into training and testing sets to evaluate the model's performance on unseen data.

Initialize and Train Random Forest Model: Create an instance of the Random Forest model and train it using the training data.

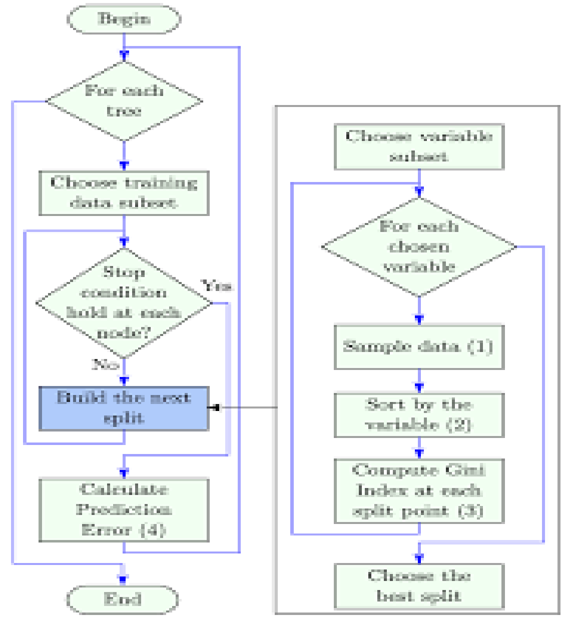


Fig 3.2: Deployment

IV. PROPOSED SYSTEM

4.1 Overview

ANN for harvest quantity and plant development involve using sophisticated machine learning models to predict crop yield and monitor plant growth stages. These neural networks leverage various data sources such as satellite imagery, weather data, soil characteristics, and historical crop yield data to make accurate predictions.

Data Exploration and Preprocessing, The code begins by importing necessary libraries and loading a dataset ('data.csv') using Pandas. Key exploratory data analysis (EDA) steps are conducted, such as checking the shape, displaying the first few rows, obtaining information on data types, and checking for missing values. The dataset primarily involves plant-related features, including growth stage, temperature, and other environmental variables. Visualizations are created to understand the distribution of temperature and the count of plants in different growth stages. Furthermore, Label Encoding is applied to convert categorical data (Plant Growth Stage) into numerical format. The dataset is then split into features (X) and the target variable (y), followed by Principal Component Analysis (PCA) to reduce dimensionality.).

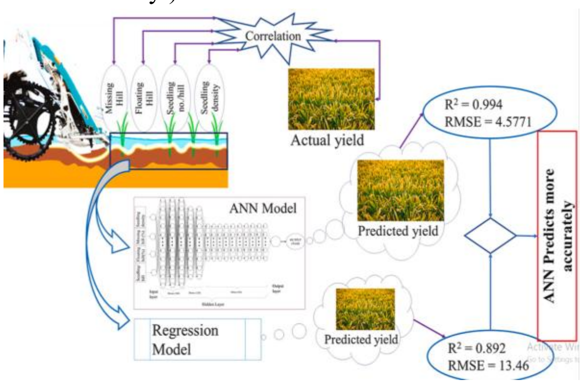


Fig 4.1: Advanced Neural Network

4.2 Block Diagram

Data Collection: Relevant data is collected from multiple sources including satellites, weather stations, IoT sensors, and historical databases. This data includes information about soil properties, weather conditions, crop types, planting dates, and past harvest yields.

Preprocessing: The collected data is preprocessed to remove noise, handle missing values, and normalize the features. This step ensures that the data is in a suitable format for training the neural network.

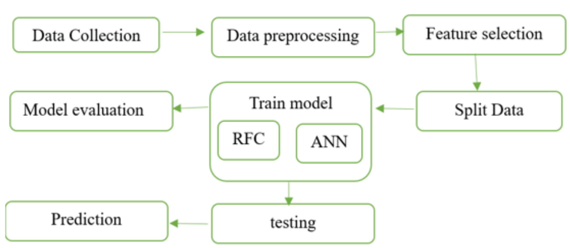


Fig 4.2: Block Diagram for Proposed methodology

4.3 Data Preprocessing

In the context of anticipating plant development and harvesting quantity in controlled greenhouse conditions using ANN , data preprocessing plays a pivotal role. The initial step involves collecting comprehensive data from various sensors monitoring factors such as temperature, humidity, light intensity, and soil moisture. Subsequent data cleaning involves addressing missing values, removing outliers, and ensuring data quality. Normalization is crucial to standardize numerical features, preventing any single variable from disproportionately influencing the model. Additionally, temporal aspects should be considered, and features was need to be engineered to capture relevant patterns in plant growth over time. Categorical data must be encoded appropriately, and the dataset should be split into training, validation, and testing sets. The final steps involve preparing the data specifically for neural networks, including proper scaling and formatting, and establishing a robust data pipeline to handle the intricacies of the model training process efficiently. Iteration and experimentation with various preprocessing techniques are essential to optimize the model's performance.

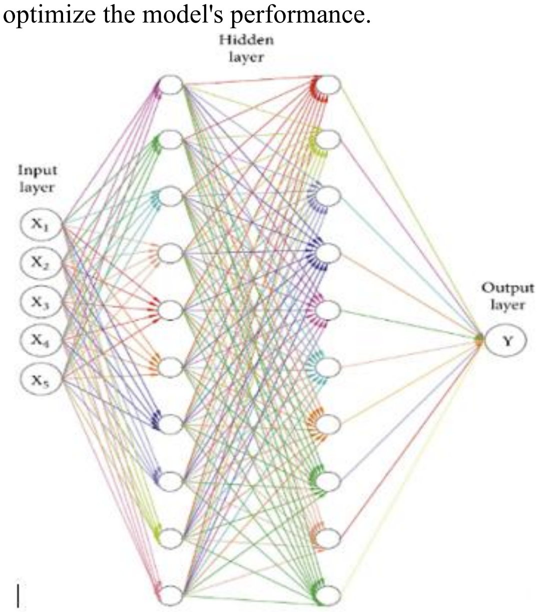


Fig 4.3: Neural networks

V UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process also be added to; or associated with, UML.

Goals: The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.

VI SOFTWARE ENVIRONMENT

6.1 What Is Python?

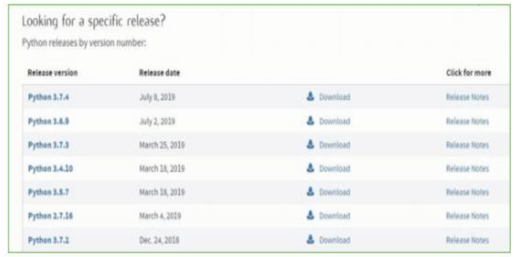
Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers must type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

6.2 Advantages of Python

Let's see how Python dominates over other languages.

Extensive Libraries: Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.



VII SYSTEM REQUIREMENTS

7.1 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

- Python IDLE 3.7 version (or)
- Anaconda 3.7 (or)
- Jupiter (or)
- Google collab

7.2 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

Operating system	:	Windows, Linux
Processor	:	minimum intel i3
Ram	:	minimum 4 GB
Hard disk	:	minimum 250GB

VIII FUNCTIONAL REQUIREMENTS

Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

Output Definition

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

IX RESULTS AND DESCRIPTION

10.1 IMPLEMENTATION AND DESCRIPTION

Data Preprocessing and Exploration, The code begins by importing necessary libraries and loading a dataset ("data.csv") into a Pandas DataFrame. The dataset's shape, basic information, and summary statistics are explored to gain insights into its structure and characteristics. Missing values are checked and found to be absent. The distribution of the "Plant Growth Stage" variable is visualized using a histogram, providing an overview of the distribution of plants across growth stages. The categorical "Plant Growth Stage" column is encoded using LabelEncoder for compatibility with machine learning models. Principal Component Analysis (PCA) is then applied to reduce the dimensionality of the dataset to 7 components, preparing the features for subsequent model training.

Model Training with Random Forest Regressor, The code splits the dataset into training and testing sets and utilizes a RandomForestRegressor for predicting the target variable (harvest quantity). The model is trained on the training set and evaluated on the test set using the R-squared metric, providing an indication of its predictive performance. A regression plot is generated to visually assess the correlation between predicted and actual values. This section demonstrates the application of a traditional machine learning algorithm to predict plant harvest quantity.

REFERENCE

[1] Jähne, Jennifer, Pham Thi Luong, Christian Blumenscheit, Andy Schneider, Jochen Blom, Pham Le Anh Minh, Ha Minh Thanh et al. "Two plant-associated Bacillus velezensis strains selected after genome analysis, metabolite profiling, and with proved biocontrol potential, were enhancing harvest yield of coffee and black pepper in large field trials." Frontiers in Plant Science 14 (2023).

[2] Mohmed, Gadelhag, Xanthea Heynes, Abdallah Naser, Weituo Sun, Katherine Hardy, Steven Grundy, and Chungui Lu. "Modelling daily plant growth response to environmental conditions in Chinese solar greenhouse using Bayesian neural network." Scientific Reports 13, no. 1 (2023): 4379.

[3] Pszczółkowski, Piotr, Barbara Sawicka, Dominika Skiba, Piotr Barbaś, and Ali Hulail Noaema. "The Use of Chlorophyll Fluorescence as an Indicator of Predicting Potato Yield, Its Dry Matter and Starch in the Conditions of Using Microbiological Preparations." Applied Sciences 13, no. 19 (2023): 10764.

[4] Cosseboom, Scott D., and Mengjun Hu. "Predicting ripe rot of grape, caused by Colletotrichum fioriniae, with leaf wetness, temperature, and the crop growth stage." PhytoFrontiers™ 3, no. 2 (2023): 303-313.

[5] Dolma, Tsering, Nawang Phuntsog, Desyong Namgail, O. P. Chaurasia, and Tsering Stobdan. "Comparing heat unit requirements for flowering and fruit harvest of cucumber in open field, shade net and greenhouse conditions." Horticulture, Environment, and Biotechnology (2023): 1-9.

[6] Mamatha, V., and J. C. Kavitha. "Machine learning based crop growth management in greenhouse environment using hydroponics farming techniques." Measurement: Sensors 25 (2023): 100665.