SOIL NUTRITION LEVEL ANALYSIS USING DEEP LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

D.LAKSHMI TEJASWINI	(21UECS0150)	**(VTU 20421)**
L.SINDHUJA	(21UECS0336)	**(VTU 19086)**
I.VARSHITHA	(21UECS0230)	**(VTU 19180)**

Under the guidance of
Mrs. U. HEMAVATHI, M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

SOIL NUTRITION LEVEL ANALYSIS USING DEEP LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

D.LAKSHMI TEJASWINI	(21UECS0150)	**(VTU 20421)**
L.SINDHUJA	(21UECS0336)	**(VTU 19086)**
I.VARSHITHA	(21UECS0230)	**(VTU 19180)**

Under the guidance of
Mrs. U. HEMAVATHI, M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled SOIL NUTRITION LEVEL ANALYSIS USING DEEP LEARNING by D.LAKSHMI TEJASWINI (21UECS0150), L.SINDHUJA (21UECS0336), I.VARSHITHA (21UECS0230) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
D.LAKSHMI TEJASWINI
Date: / /

(Signature)
L.SINDHUJA
Date: / /

(Signature)
I.VARSHITHA
Date: / /

APPROVAL SHEET

This project report entitled SOIL NUTRITION LEVEL ANALYSIS USING DEEP LEARNING by D.LAKSHMI TEJASWINI (21UECS0150), L.SINDHUJA (21UECS0336), I.VARSHITHA (21UECS0230) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	Mrs. U. Hemavathi, M.E.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mrs. U. HEMAVATHI, M.E.,** for her cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U. HEMAVATHI, M.E, Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

D.LAKSHMI TEJASWINI (21UECS0150)
L.SINDHUJA (21UECS0336)
I.VARSHITHA (21UECS0230)

ABSTRACT

The Crop Recommendation System represents a comprehensive solution for modern agriculture by integrating Convolutional Neural Networks (CNNs) into a user-friendly web application. This innovative project leverages machine learning techniques to predict suitable cropsbased on environmental and soil conditions, providing farmers with valuable insights for optimized cultivation practices. The project initiates with the exploration and preprocessing of an agricultural dataset obtained from an official crop database. Data cleaning procedures handle duplicates and missing values, while Exploratory Data Analysis (EDA) offers insights into the distribution and relationships among critical agricultural parameters such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall. Unlike traditional machine learning models, this project employs a Convolutional Neural Network (CNN) implemented using the Keras library. The CNN architecture allows the model to learn hierarchical representations from input images, in this case, the agricultural parameters. The model is trained to recognize patterns that indicate optimal crop recommendations based on the input conditions. The heart of the project is a user-friendly web interface developed using Flask, HTML, and CSS. The web application prompts users to input specific agricultural parameters, such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall. The input values are then processed and fed into the CNN model for prediction.The trained CNN model, along with the required preprocessing components (StandardScaler and LabelEncoder), is saved and seamlessly integrated into the Flask web application. Farmers interact with the system by entering agricultural parameters through a user-friendly web form. The CNN model processes these inputs, extracting meaningful features and making predictions about the optimal crop for cultivation. The results are displayed on the website, offering farmers actionable insights for decision-making. The Crop Recommendation System with CNN integration signifies a paradigm shift in agriculture, bringing advanced neural network techniques to the forefront of crop prediction. By combining the power of CNNs with a user-friendly interface, this project strives to empower farmers with accurate and efficient recommendations, fostering sustainable and optimized agricultural practices.

Keywords:

Agriculture, Crop Recommendation, Convolutional Neural Network, Flask, Machine Learning, Predictive, Web Application

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AP	Andhra Pradesh
CNNs	Convolutional Neural Networks
CSS	Cascading Style Sheet
EDA	Exploratory Data AnalysiS
GPS	Global Positioning System
HTML	Hyper Text Markup Language
KNN	K-Nearest Neighbour
PH	Potential of Hydrogen
PA	Precision Agriculture
RNNs	Recurrent Neural Networks
RIO	Return On Investment
ML	Mechine Learning
TCI	Temperature Condition Inde
UI	User Interface
UML	Unified Modeling Language

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The Soil Nutrition Level Analysis using Deep Learning is an innovative and intelligent application designed to revolutionize modern agriculture by harnessing the power of Convolutional Neural Networks (CNNs) for crop prediction. Agriculture, a critical sector for global sustenance, faces challenges in optimizing crop selection based on diverse environmental and soil conditions. This project addresses these challenges by combining data science, machine learning, and web development to provide farmers with accurate and tailored recommendations for crop cultivation. Traditional crop recommendation systems often rely on conventional machine learning models. However, this project adopts a novel approach by implementing a CNN, a specialized neural network architecture renowned for its prowess in image recognition tasks. By treating agricultural parameters as an image, the CNN can capture intricate patterns and relationships among the variables, ultimately enhancing the accuracy of crop predictions. The project begins with an exploration of an extensive agricultural dataset sourced from an official crop database. Rigorous preprocessing techniques, including data cleaning and Exploratory Data Analysis (EDA), set the foundation for a robust model. The integration of a CNN into the system elevates its predictive capabilities, allowing for nuanced analysis of input parameters to generate precise and informed crop recommendations. To facilitate user interaction, a user-friendly web application is developed using Flask, HTML, and CSS. The application prompts users to input key agricultural parameters such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall. The input data is processed, scaled, and seamlessly fed into the CNN model, which dynamically evaluates the conditions and predicts the most suitable crop for cultivation. The Crop Recommendation System not only showcases the potential of advanced neural network architectures in agriculture but also emphasizes the importance of accessible and practical tools for farmers. By providing actionable insights through a user-friendly interface, this project strives to empower farmers to make informed decisions, fostering sustainable and optimized agricultural practices in an everevolving landscape.

1.2 Aim of the project

The primary aim of the Soil Nutrition Level Analysis using Deep Learning is to revolutionize and modernize agricultural practices by integrating advanced machine learning, specifically Convolutional Neural Networks (CNNs), into a user-friendly web application. The project seeks to address the challenges faced by farmers in selecting the most suitable crops based on diverse environmental and soil conditions, ultimately enhancing the efficiency and sustainability of crop cultivation.

1.3 Project Domain

Project Domain: Precision Agriculture and Crop Management

The Soil Nutrition Level Analysis using Deep Learning operates within the domain of Precision Agriculture and Crop Management. Precision Agriculture, also known as precision farming, involves the use of advanced technologies to optimize crop yields, reduce waste, and enhance overall farm efficiency. In this context, the project focuses on leveraging data science and machine learning techniques to provide farmers with precise recommendations for crop cultivation.

Key Components of the Project Domain:

Data-Driven Decision Making:

Harnessing data to make informed decisions in agriculture. Utilizing machine learning models to analyze and interpret complex datasets related to soil and environmental conditions.

Crop Prediction Models:

Implementing advanced machine learning models, such as Convolutional Neural Networks (CNNs), for accurate and nuanced crop predictions.Enhancing the prediction capabilities by treating agricultural parameters as image-like data, allowing for the extraction of intricate patterns.

Environmental and Soil Factors:

Considering key environmental factors, including temperature, humidity, and rainfall. Analyzing soil composition parameters such as nitrogen, phosphorus, potassium, and pH.

Web-Based User Interface:

Developing a user-friendly web application for seamless interaction with the recommendation system. Allowing farmers to input specific agricultural parameters through an intuitive interface.

Precision Agriculture Technologies:

Aligning with the broader field of precision agriculture, which integrates technologies such as sensors, GPS guidance, and automated machinery. Positioning the project as a component of the evolving landscape of precision agriculture technologies.

Optimized Crop Cultivation:

Empowering farmers to make data-driven decisions leading to optimized crop cultivation. Enhancing overall farm productivity and profitability through targeted and efficient farming practices. The project operates at the intersection of agriculture, data science, and machine learning, contributing to the evolution of modern farming practices. By focusing on precision agriculture and crop management, the **Soil Nutrition Level Analysis using Deep Learning** aims to be a valuable tool for farmers seeking to enhance their decision-making processes and achieve sustainable and efficient crop cultivation.

1.4 Scope of the Project

Scope of the Project:

The Soil Nutrition Level Analysis using Deep Learning envisions a comprehensive scope within the domain of Precision Agriculture, aiming to provide farmers with advanced tools for optimized crop selection.

Crop Recommendations Based on Environmental Parameters:

The system analyzes crucial environmental parameters such as temperature, humidity, and rainfall to recommend crops suited to specific climatic conditions.

Soil Composition Analysis:

Incorporating soil composition factors including nitrogen, phosphorus, potassium, and pH to tailor recommendations based on soil health and nutrient levels.

Integration of Convolutional Neural Networks (CNNs):

Leveraging CNNs to process agricultural parameters as image-like data, allowing the model to capture intricate patterns and relationships for more accurate predictions.

Web-Based User Interface:

Developing an intuitive web interface that enables farmers to input agricultural parameters seamlessly, making the system accessible to users with varying technical backgrounds.

Dynamic and Real-time Predictions:

Providing real-time predictions for crop recommendations, ensuring that farmers receive timely and relevant insights for decision-making.

Model Training and Evaluation:

Rigorously training the machine learning model with a diverse dataset and evaluating its performance to ensure reliable and accurate crop predictions.

User-Friendly Output:

Presenting the recommendations in a user-friendly format, allowing farmers to easily interpret and implement the suggested crop cultivation strategies.

Adaptability to Diverse Agricultural Practices:

Designing the system to accommodate a variety of crops and agricultural practices, ensuring its applicability across different regions and farming methods.

Educational and Training Resources:

Providing educational resources within the system to empower users with knowledge on the factors influencing crop recommendations and fostering a better understanding of precision agriculture.

Compatibility with Existing Technologies:

Ensuring compatibility with existing precision agriculture technologies and allowing for seamless integration with other farm management systems. The scope of the Crop Recommendation System extends beyond conventional crop recommendation approaches, aiming to introduce innovative technologies and methodologies for sustainable and datadriven agriculture. As precision agriculture continues to evolve, the project stands as a valuable tool to assist farmers in making informed decisions for optimized crop cultivation.

Chapter 2

LITERATURE REVIEW

[1] A. Motwani (2022) india, renowned for its agricultural prominence, stands among the top global producers for various crops. Despite the centrality of the Indian farmer to the agricultural sector, many of them occupy a lower socio-economic status. Additionally, determining the most suitable and profitable crop for a particular soil remains a challenge, given the diverse soil types across geographical regions. This study introduces a crop recommendation system that utilizes a Convolutional Neural Network (CNN) and a Random Forest Model to predict the ideal crop based on factors such as region, soil type, yield, selling price, and more. The CNN achieved an accuracy of 95.21Algorithm demonstrated an accuracy of 75 [2].

[2] G. Chauhan (2021) states that Precision agriculture involves integrating the latest technology into farming practices. The agricultural sector generates vast amounts of data, and various data mining techniques are applied to optimize its utilization. This paper explores different classification algorithms within the field of data mining, specifically focusing on their application to a dataset accumulated over several years for predicting soybean crop yield. Subsequently, a comparative analysis is conducted to determine the most effective classification algorithm for accurate yield prediction within the realm of classification techniques[5].

[3] M. Paul (2020) states that effective agriculture planning is crucial for the economic growth and food security of agrarian nations. The selection of crops is a key consideration in this planning process, influenced by factors such as production rates, market prices, and government policies. Previous research has delved into yield rate prediction, weather forecasting, soil and crop classification using statistical methods or machine learning techniques to inform agricultural planning decisions. When faced with the challenge of choosing among multiple crop options within limited land resources, the crop selection becomes a complex puzzle[7].

[4] R. Kumar (2022) states that the production of agricultural crops is influenced by a range of factors encompassing biology, climate, economy, and geography. These factors have varied impacts on agriculture, and their effects can be measured using suitable statistical methodologies. By applying such methodologies and techniques to historical crop yield data, valuable information and insights can be derived. This knowledge proves beneficial for farmers and government organizations in making informed decisions and formulating policies aimed at enhancing production. This paper specifically concentrates on utilizing data mining techniques to extract knowledge from agricultural data, with the goal of estimating crop yields for major cereal crops in key districts of Bangladesh[8].

[5] S. Pudumalar (2022) states that Precision Agriculture (PA) initially emerged to tackle the variability in soil and crop parameters on a large scale within developed nations. However, the fundamental principles of PA can be adapted for farm-based agriculture, catering to small and marginal farmers in Developing Countries. This approach involves creating a farmer-soilcrop database from field data, incorporating crop calendars provided by agricultural experts, real-time collection of parameters like temperature and rainfall through sensors, and employing an analytical model to simulate the crop calendar using static, semi-static, and dynamic inputs. This leads to the delivery of support advisories at the farmer and crop levels, accessible through devices such as mobile phones and tablets[10].

[6] Y. Sanghvi (2019) states that Precision agriculture involves incorporating the latest technology into farming practices. The agricultural sector generates a substantial amount of data, and various data mining techniques are applied to optimize its utilization. This paper explores different algorithms associated with data mining classification techniques. These algorithms are applied to a dataset collected over several years to predict the yield of soybean crops. Additionally, a comparative analysis is conducted to identify the most suitable classification algorithm for accurate yield prediction in the context of classification techniques[11].

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing agricultural system in many regions relies heavily on traditional and manual practices for crop selection. Farmers often make decisions based on their experience, local knowledge, and sometimes limited advice from agricultural experts. The traditional approach may involve trial and error, leading to suboptimal crop choices and reduced overall productivity. Additionally, external factors such as changing climate conditions, varying soil types, and unpredictable weather patterns make it challenging for farmers to consistently make informed decisions about the crops they cultivate.

In some cases, farmers may seek guidance from agricultural extension services, which provide recommendations based on general guidelines and historical data. However, these recommendations may not always consider the specific and dynamic conditions of individual farms, leading to less accurate and personalized suggestions.

The limitations of the existing system highlight the need for a more advanced and data-driven approach to crop recommendation. Integrating modern technologies, such as data mining, machine learning, and precision agriculture, can offer more precise and tailored recommendations for farmers, taking into account various environmental and soil factors to optimize crop selection and enhance overall agricultural productivity.

The limitations of the existing system highlight the need for a more advanced and data-driven approach to crop recommendation. Integrating modern technologies, such as data mining, machine learning, and precision agriculture, can offer more precise and tailored recommendations for farmers, taking into account various environmental and soil factors to optimize crop selection and enhance overall agricultural productivity.

3.2 Proposed System

The proposed system introduces an innovative approach to crop recommendation, leveraging advanced technologies such as data mining, machine learning, and precision agriculture to address the limitations of traditional farming practices. The primary goal is to provide farmers with accurate and personalized recommendations for crop selection based on site-specific parameters, ultimately enhancing agricultural productivity.

The system incorporates a comprehensive ensemble model that utilizes machine learning algorithms, including Random Tree, CHAID, KNearest Neighbor, and Naive Bayes. By employing a majority voting technique, the system ensures robust and reliable crop recommendations, taking into account a wide range of factors such as soil characteristics, soil types, and historical crop yield data.

Precision agriculture techniques play a pivotal role in the proposed system, allowing for the collection and analysis of real-time data from the field. This includes factors like temperature, rainfall, humidity, and other environmental variables, contributing to more accurate predictions and personalized recommendations. The integration of these technologies aims to empower farmers with timely and datadriven insights, enabling them to make informed decisions about crop selection, leading to increased yields and improved overall agricultural outcomes.

In summary, the proposed system represents a significant advancement over the existing practices by harnessing the power of modern technologies to create a sophisticated and adaptive crop recommendation system tailored to the unique conditions of each farming site.

3.3 Feasibility Study

The feasibility study for the proposed crop recommendation system involves a comprehensive assessment of various aspects to determine the practicality and viability of implementing the system. The study encompasses several key dimensions:

3.3.1 Economic Feasibility:

Cost-Benefit Analysis: Conducting a detailed analysis of the costs associated with system development, implementation, and maintenance against the anticipated benefits, including increased crop yield and financial returns for farmers. Return on Investment (ROI): Calculating the projected ROI to determine the economic viability of the proposed system.

3.3.2 Technical Feasibility

System Architecture: Assessing the technical requirements and architecture needed for the implementation of the ensemble model, precision agriculture techniques, and data mining algorithms. Data Availability: Evaluating the availability and accessibility of relevant data, including historical crop yield data, soil characteristics, and real-time environmental variables.

3.3.3 Social Feasibility

User Acceptance: Gauging the willingness and acceptance of farmers to adopt and use the proposed system, considering factors such as userfriendliness and accessibility. Training Requirements: Identifying the training needs for farmers and stakeholders to effectively use the system.

3.4 System Specification

If We use a certain project in an efficient way, we need hardware components and software components to be present in a computer. These components are been used as a guideline for a project. While the Increase Of higher processing power and new versions of software it increases time management. By this, we can conclude that it plays a bigger role in computer systems.

3.4.1 Hardware Specification

- Processor : I3/I5 Intel Processor
- RAM : 8 GB

3.4.2 Software Specification

- PYCHARM
- PYTHON
- PRE-TRAINED MODELS
- DEEP LEARNING MODELS
- WEBCAM

3.4.3 Standards and Policies

Anaconda Prompt

Anaconda prompt is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 Soil Nutrition Level Analysis Architecture

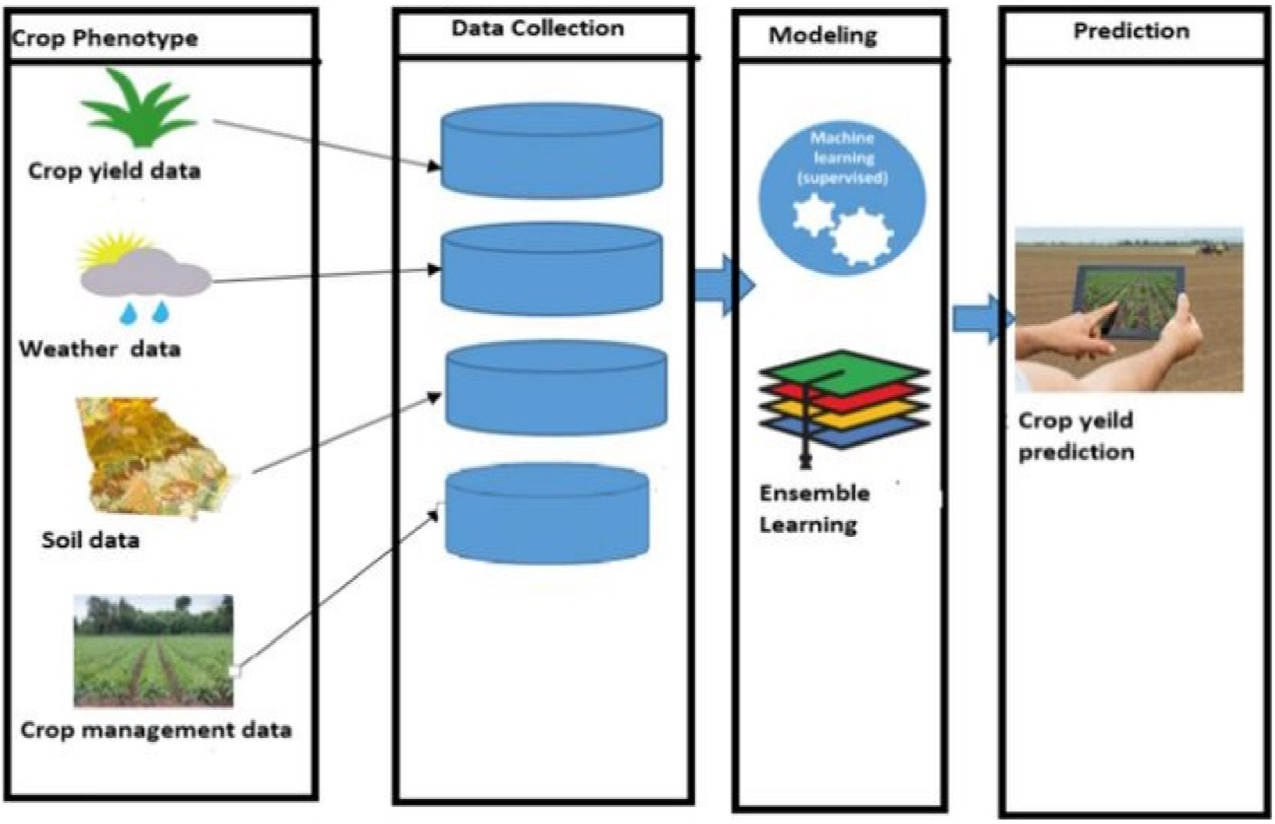


Figure 4.1: **General Architecture of Soil Nutrition Level Analysis using Deep Learning**

In the above Figure 4.1 Crop prediction involves using various technologies and methodologies to forecast the yield and growth of crops. The crop yeild data, weather data, soil data should be collected. The general architecture of Soil Nutrition Level Analysis using Deep Learning for crop prediction can be a combination of data acquisition, data pre-processing, feature extraction, model training, and prediction. The data pre-processing involves correcting inaccurate data from the dataset.

4.2 Design Phase

4.2.1 Data Flow Diagram

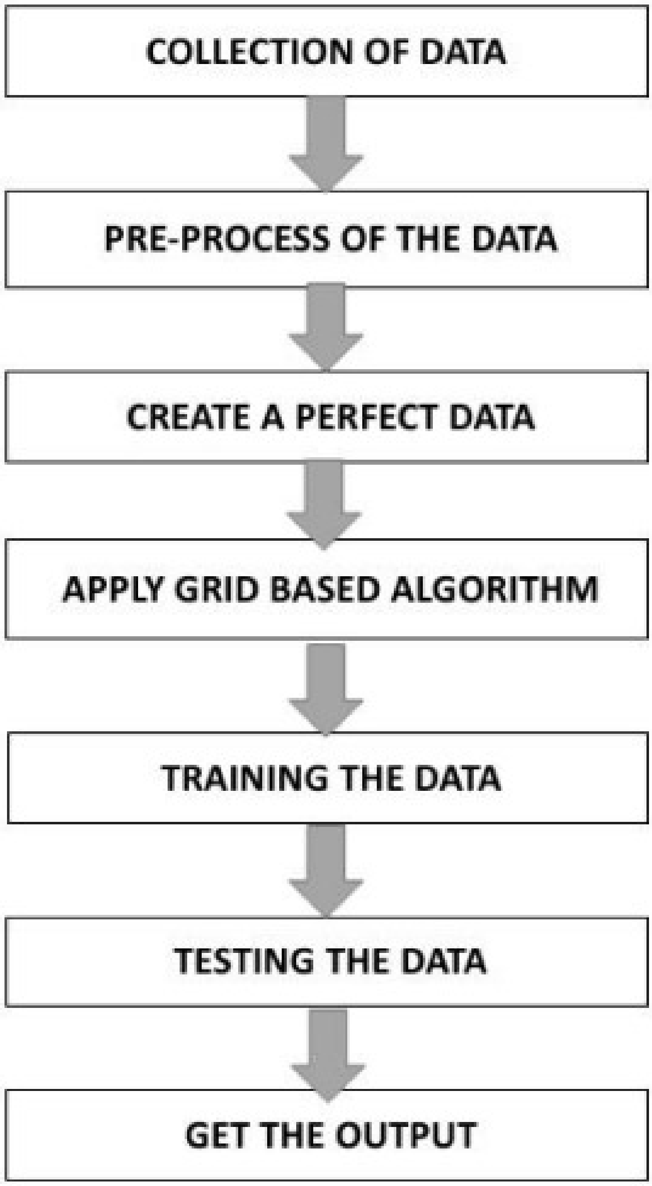


Figure 4.2: **Data Flow Diagram of Soil Nutrition Level Analysis using Deep Learning**

In the above Figure 4.2 data flow diagram of soil nutrition level analysis using deep learning represents structure of the data. Data flow diagram is a type of diagram in the Unified Modeling Language (UML) that represents the structure of a system by showing the data, their attributes, operations, and the relationships among them. For a crop prediction system, the data flow diagram contains data, models, users, and more.

4.2.2 Use Case Diagram

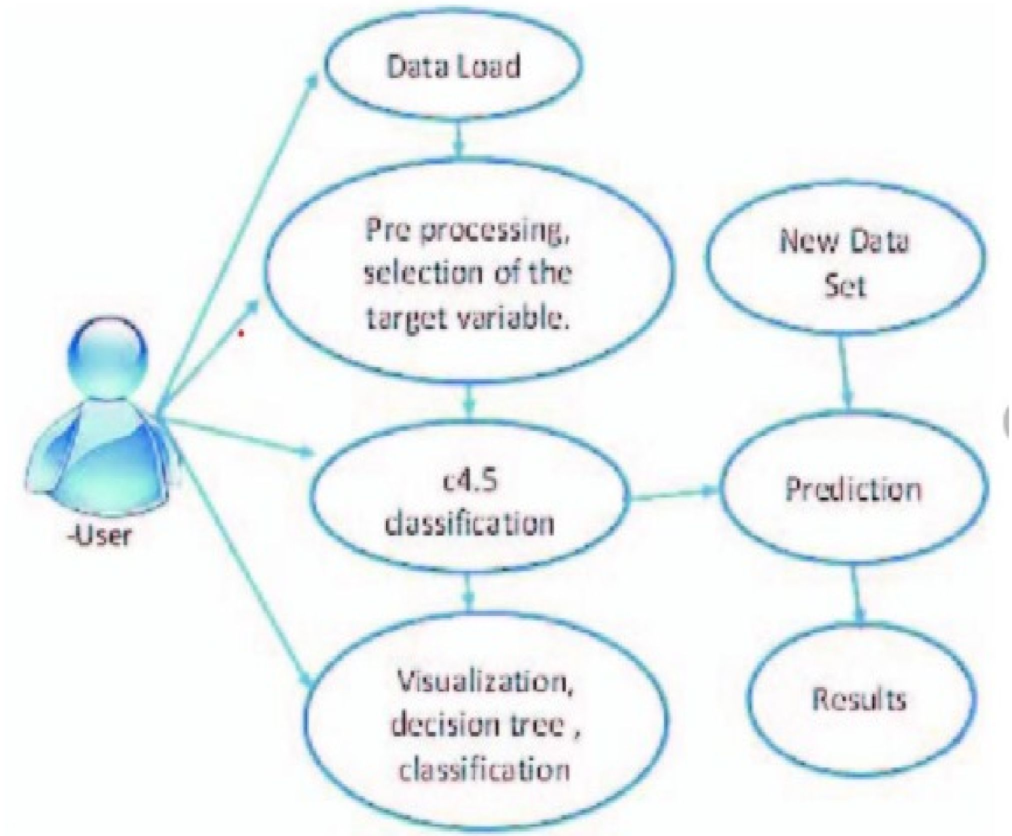


Figure 4.3: **Use Case diagram of Soil Nutrition Level Analysis using Deep Learning**

In the above Figure 4.3 use case for crop prediction outlines a specific scenario in which a crop prediction system is utilized to achieve a particular goal. Maximize crop yield by leveraging a crop prediction system to make informed decisions. The crop prediction system is installed and configured. The classification and visualization are related to the user in the Use Case diagram. The user classifies the given data using classification and predicts the data. Through the prediction the result will be dispalyed.

4.2.3 Class Diagram

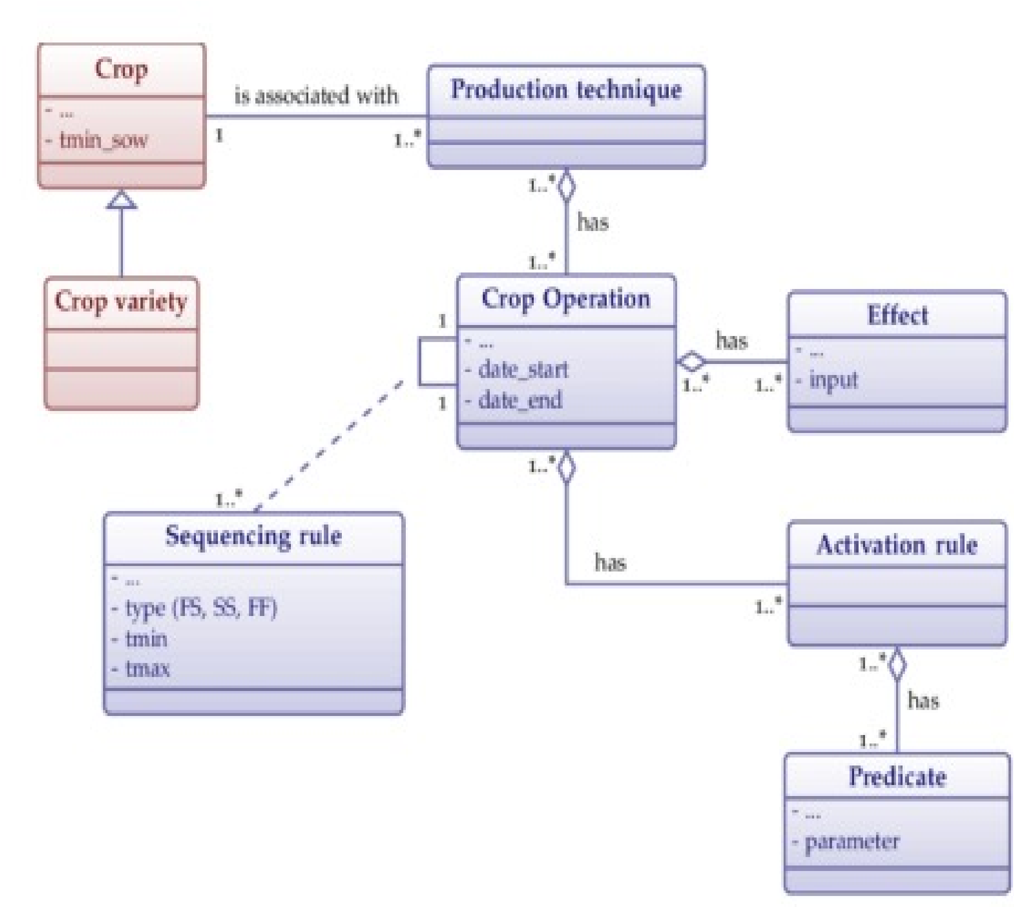


Figure 4.4: **Class Diagram of Soil Nutrition Level Analysis using Deep Learning**

In the above Figure 4.4 class diagram of soil nutrition level analysis using deep learning represents the main class that encapsulates the entire crop prediction system. It contains instances of Weather Data, Crop Data, and Prediction Model. Weather Data represents the class containing weather-related data such as temperature, humidity, precipitation, and wind speed. Crop Data represents the class containing crop-related data such as crop type, soil moisture, and nutrient levels. The class diagram contains the crop, crop variety, crop operation, predicate, effect, sequencing rule, activation rule which were connected with each other.

4.2.4 Sequence Diagram of Soil Nutrition Level Analysis using Deep Learning

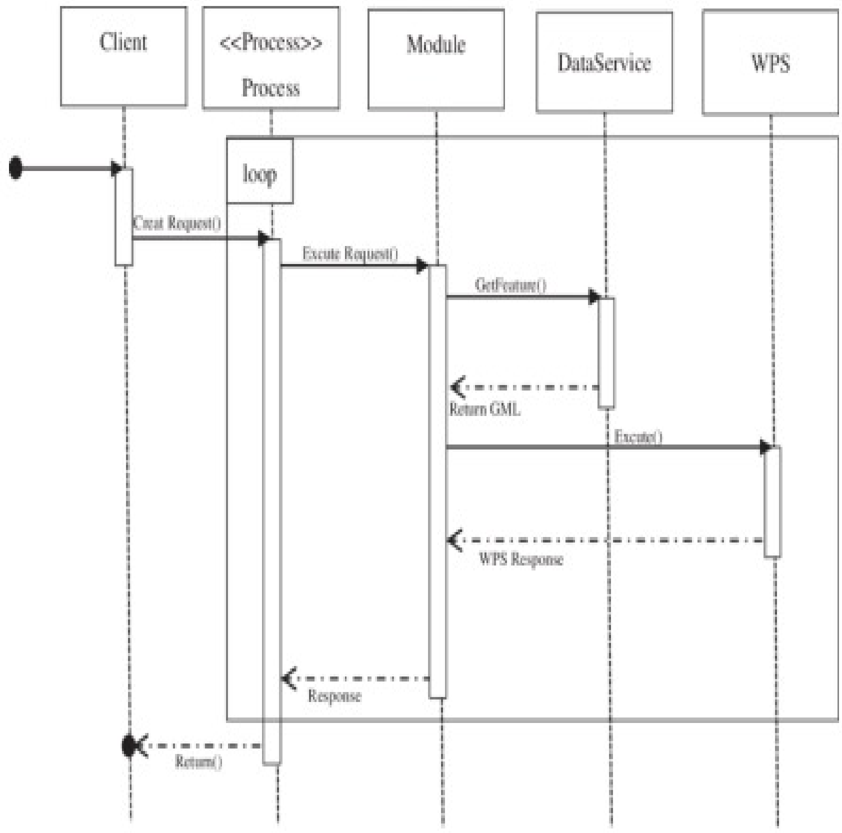


Figure 4.5: **Sequence Diagram**

In the above Figure 4.5 Sequence diagram of Soil Nutrition Level Analysis using Deep Learning illustrates the interactions and flow of messages between different components or objects involved in the crop prediction process. The Farmer interacts with the Soil Nutrition Level Analysis using Deep Learning to initiate the crop prediction process. The Farmer gathers data and inputs it into the Soil Nutrition Level Analysis using Deep Learning. The Soil Nutrition Level Analysis using Deep Learning analyzes the data, generates predictions, and displays the results. The Soil Nutrition Level Analysis using Deep Learning may also provide recommendations based on the predictions.

4.2.5 Collaboration diagram

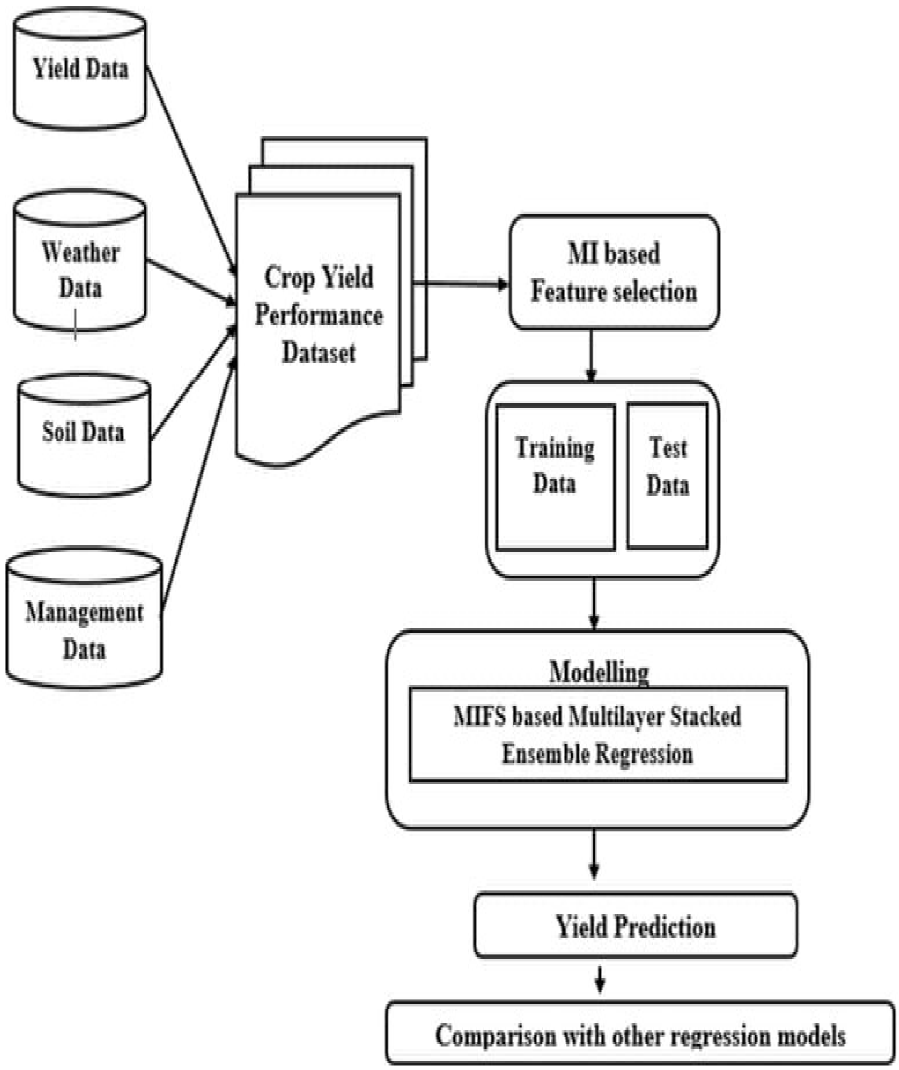


Figure 4.6: **Collaboration diagram of Soil Nutrition Level Analysis using Deep Learning**

In the above Figure 4.6 the Farmer initiates the process by interacting with the Soil Nutrition Level Analysis using Deep Learning. The Farmer communicates with the Soil Nutrition Level Analysis using Deep Learning by calling methods such as gather Data,input Data,system Analysis,generate Predictions, display Results, and display Recommendations. The Soil Nutrition Level Analysis using Deep Learning interacts with its internal components, namely Weather Data and Prediction Model, to perform various tasks. The Weather Data and Prediction Model objects provide the necessary data and predictions for the crop prediction process.

4.2.6 Activity Diagram

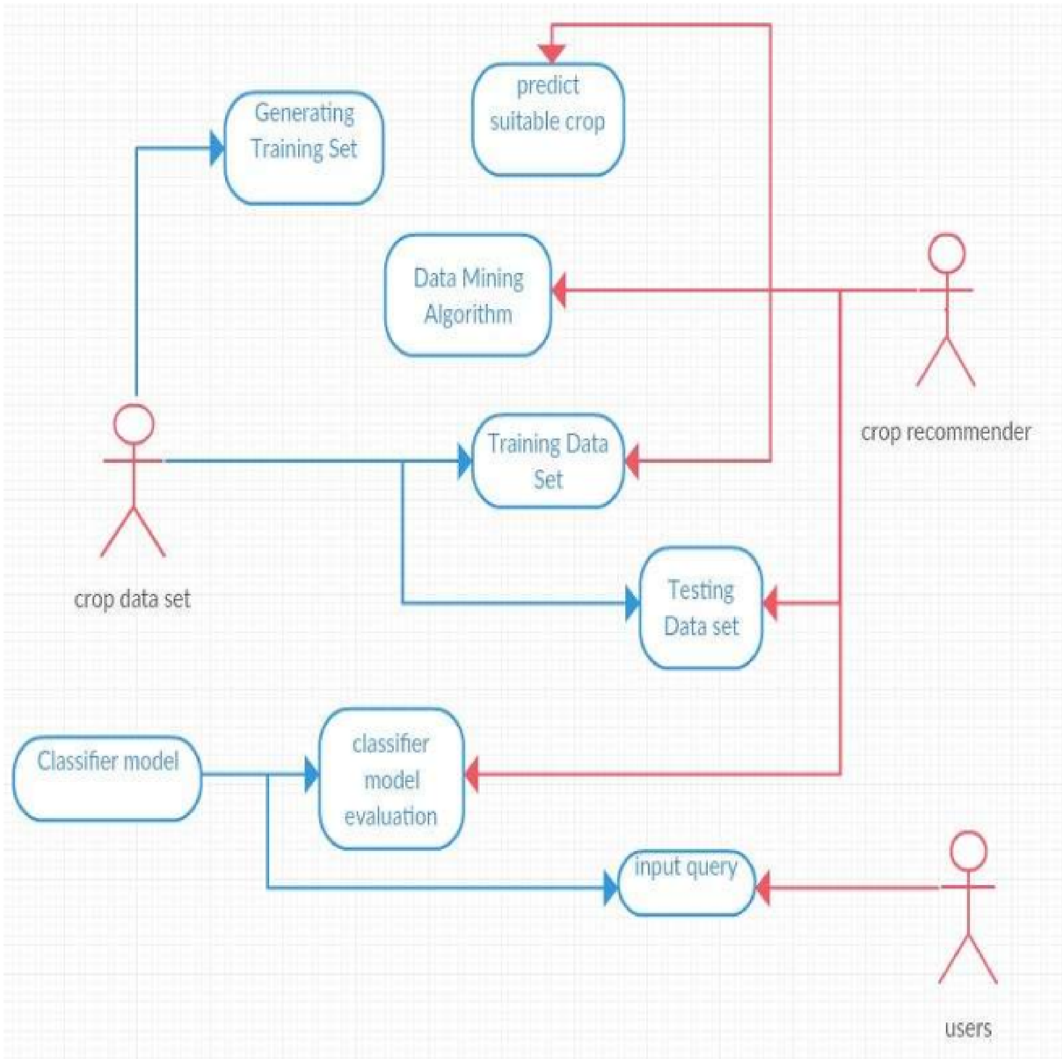


Figure 4.7: **Activity Diagram of Nutrition Level Analysis using Deep Learning**

In the above Figure 4.7 Start Process (Initial Node): The process begins with the initiation of the crop prediction system.Gather Weather Data (Activity): The system collects relevant weather data, including temperature, humidity, precipitation, and wind speed.Gather Crop Data (Activity): The system collects crop-related data, such as crop type, soil moisture, and nutrient levels. Preprocess Data (Activity): Data preprocessing activities are performed, including cleaning, normalization, and handling missing values.

4.3 Algorithm & Pseudo Code

4.3.1 Convolutional Neural Network Algorithm

1. Convolutional Neural Network (CNN): Input layer: Receive input images. Convolutional layers Detect features through convolution operations. Activation function Introduce non-linearity (e.g., ReLU). Pooling layers Downsample feature maps. Fully connected layers Classify features. Output layer Generate predictions.

2. Random Forest: Randomly select a subset of features. Construct a decision tree based on the selected features. Repeat the process to create multiple trees. Ensemble the trees to make predictions by averaging or voting.

3. Ensemble Model with Majority Voting: Train multiple models using different algorithms or subsets of data. Make predictions with each model. Aggregate predictions using majority voting. Output the class with the most votes as the final prediction.

4. Data Preprocessing: Data cleaning Handle missing values and outliers. Data normalization Scale features to a similar range. Data encoding Convert categorical variables into numerical representations (e.g., one-hot encoding). Data splitting Divide data into training and testing sets.

5. Training and Testing: Split data into training and testing sets. Train the model on the training set using appropriate algorithms. Validate the model's performance on the testing set using evaluation metrics (e.g., accuracy, precision, recall).

6. Prediction: Use the trained model to make predictions on new or unseen data. Output the predicted classes or values.

4.3.2 Pseudo Code

```
    # Import necessary libraries and modules
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```python
# Load dataset
dataset = pd.read_csv('crop_dataset.csv')
# Data preprocessing
# Handle missing values, remove duplicates
dataset = dataset.drop_duplicates().dropna()
# Define features (X) and target variable (y)
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
# Encode categorical labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# Train Random Forest model
random_forest = RandomForestClassifier(n_estimators=100,
random_state=42)
random_forest.fit(X_train, y_train)
# Evaluate model on the test set
accuracy = random_forest.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy}")
# User input for prediction
new_input = np.array([24, 59, 18, 23.70, 54.21, 6.64, 124.77])
new_input_scaled = scaler.transform(new_input.reshape(1, -1))
# Make prediction
prediction = random_forest.predict(new_input_scaled)
predicted_crop = label_encoder.inverse_transform(prediction)
print(f"Predicted crop: {predicted_crop[0]}")
```

4.4 Module Description

4.4.1 Convolutional Neural Networks

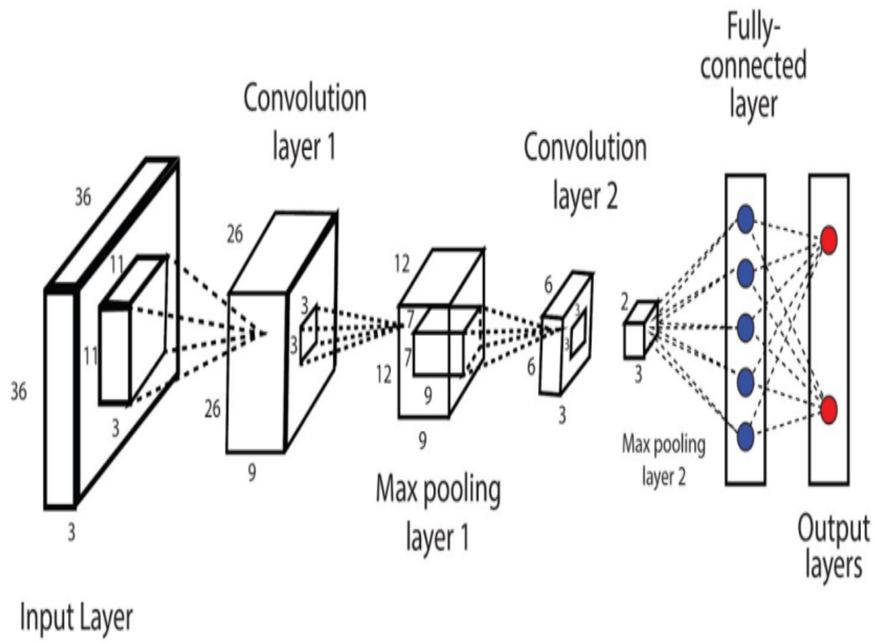


Figure 4.8: **Convolutional Neural Networks**

In the above Figure 4.8 A Convolutional Neural Network (CNN) tree diagram illustrates the hierarchical structure of its layers. The input layer, often an image, initiates the flow, followed by convolutional layers that employ filters to extract features. Activation layers introduce non-linearities like ReLU, and pooling layers reduce spatial dimensions. Fully connected layers connect neurons across layers, leading to the output layer. Flattening layers may be present to convert convolutional outputs into vectors. The interconnected nodes and arrows depict the flow of information, showcasing how the network processes input data through convolutions, non-linearities, and pooling to generate its final output.

4.4.2 Collecting and Preprocessing Module

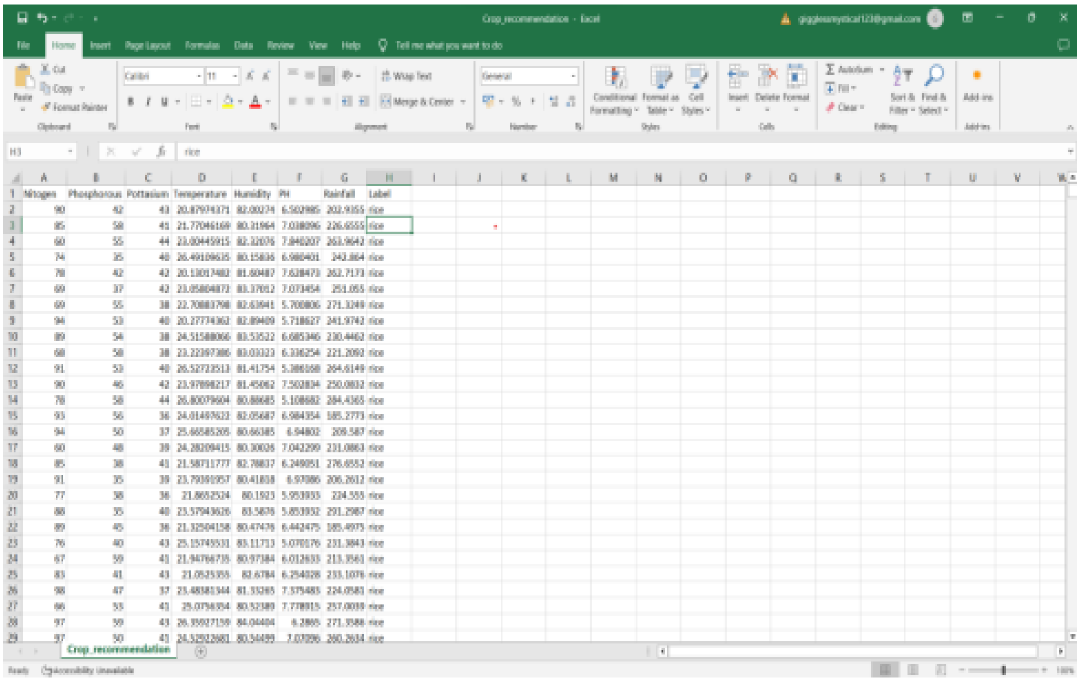


Figure 4.9: **Collecting and Preprocessing**

In the above Figure 4.9 Collecting and preprocessing data is a critical phase in machine learning workflows. The data collection involves gathering relevant information for the task at hand, whether it be images, text, or numerical data. Once collected, preprocessing steps are applied to enhance the data's quality and usability. This includes handling missing values, scaling numerical features, encoding categorical variables, and addressing outliers. Data normalization and standardization may also be performed to ensure consistency. Additionally, techniques like data augmentation might be applied, especially in image-related tasks, to increase the diversity of the dataset. Overall, the meticulous process of collecting and preprocessing data lays the foundation for building effective and robust machine learning models by ensuring the data is representative, clean, and appropriately formatted for analysis and training.

4.4.3 Model Training Module

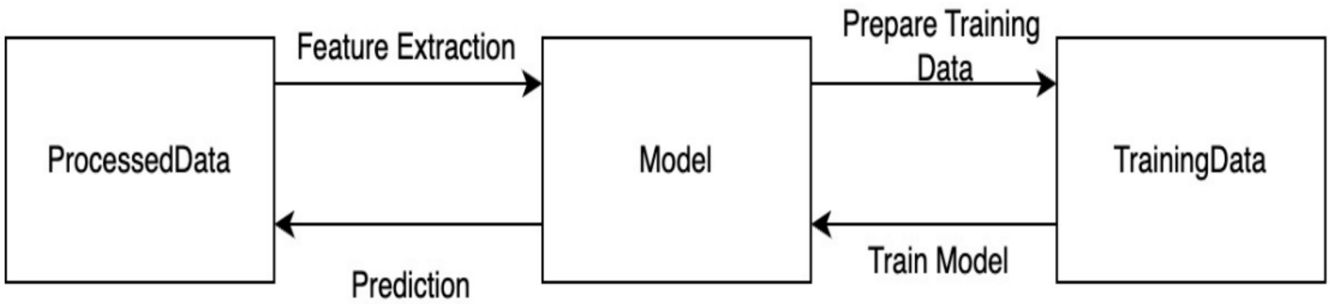


Figure 4.10: **Model Training Module**

In the above Figure 4.10 The dataset is split into training and testing sets using the functions. The training set is used to train the machine learning models, and the testing set is used to evaluate their performance. The models are compiled with appropriate loss functions and optimizers, and the training involves adjusting the model parameters to minimize the loss.

4.4.4 Prediction Module

After training, the models are used to make predictions on new input data. For the web-based application, when a user inputs soil and climate parameters, the models process the data and predict the most suitable crop for cultivation.

4.5 Steps to execute/run/implement the project

4.5.1 Step1

1. Problem Understanding and Requirement Analysis:

- Clearly define the problem: Helping farmers in crop selection based on environmental factors.
- Understand user requirements: Input features (nitrogen, phosphorus, etc.), desired output (recommended crop).

2. Data Collection:

- Gather a comprehensive dataset containing information on various crops and environmental factors.

- Ensure the dataset is diverse, representative, and includes relevant features.

3. Data Preprocessing:

- Handle missing values, duplicate entries, and outliers. Encode categorical variables (e.g., crop labels) using Label Encoding. Split the dataset into features (X) and target variable (y).

4.5.2 Step2

1. Exploratory Data Analysis (EDA):

- Visualize data distribution and relationships using plots and graphs. Understand correlations between different environmental factors.
- Identify patterns that may influence crop recommendations.

2. Model Selection:

- Split the dataset into training and testing sets.
- Standardize or normalize numerical features.
- Train the selected models using the training dataset.

3. Model Evaluation:

- Develop a Flask web application (app.py) for user interaction.
- Load the trained model, scaler, and label encoder in the Flask application.

4.5.3 Step3

1. User Interface (UI) Design:

- Design a user-friendly interface for users to input environmental factors.
- Implement a ”Predict” button to trigger the recommendation process.

2. Deployment:

- Choose a hosting platform (e.g., Heroku, AWS) to deploy the Flask application.
- Ensure all dependencies are satisfied on the deployment platform

3. Testing:

- Test the web application thoroughly, including input validation and error handling.
- Verify that the model provides accurate crop recommendations.

4. Future Improvements:

- Identify potential enhancements, such as incorporating more features or expanding the dataset.
- Consider continuous model training to adapt to changing agricultural conditions.

5. Reporting:

- Create a comprehensive report summarizing the project, methodologies, results, and potential future work.
- Following these steps should help you successfully execute the crop recommendation system project.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

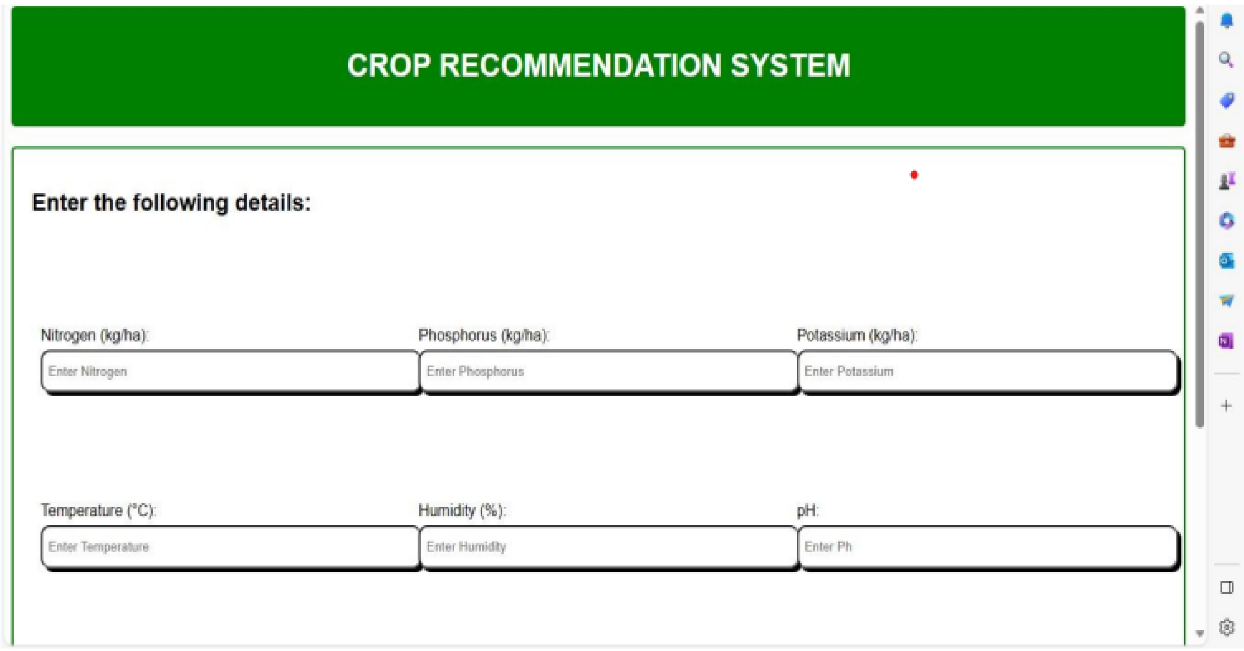


Figure 5.1: **Input Design**

In the above Figure 5.1 Designing an input system for crop recommendation involves gathering relevant information about the agricultural environment, soil conditions, climate, and other factors that affect crop growth.

5.1.2 Output Design

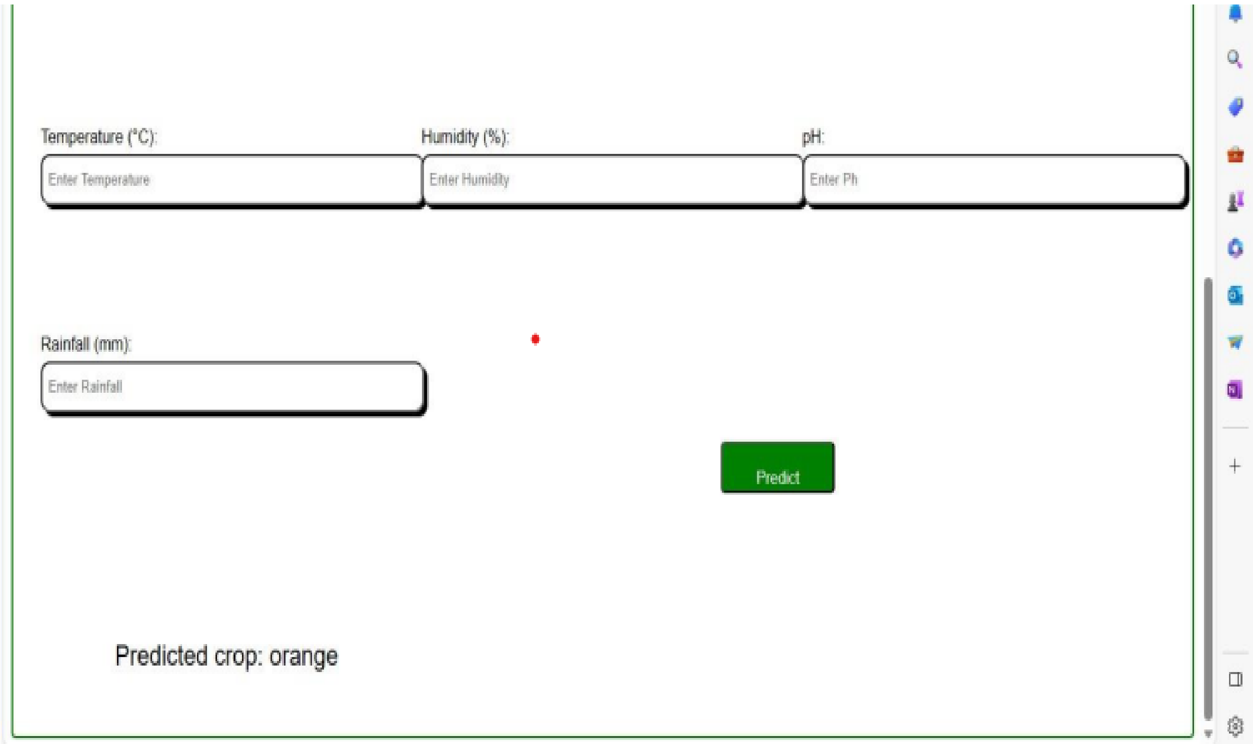


Figure 5.2: **Output Design**

In the above Figure 5.2 The crop recommendation output is presented through a user-friendly interface, offering farmers a clear and concise list of recommended crops based on comprehensive input parameters. Each recommended crop is accompanied by detailed information, including growth requirements, optimal planting seasons, and estimated yields. The crop is predicted by also using some parameters such as humidity, rainfall.

5.1.3 Test Result

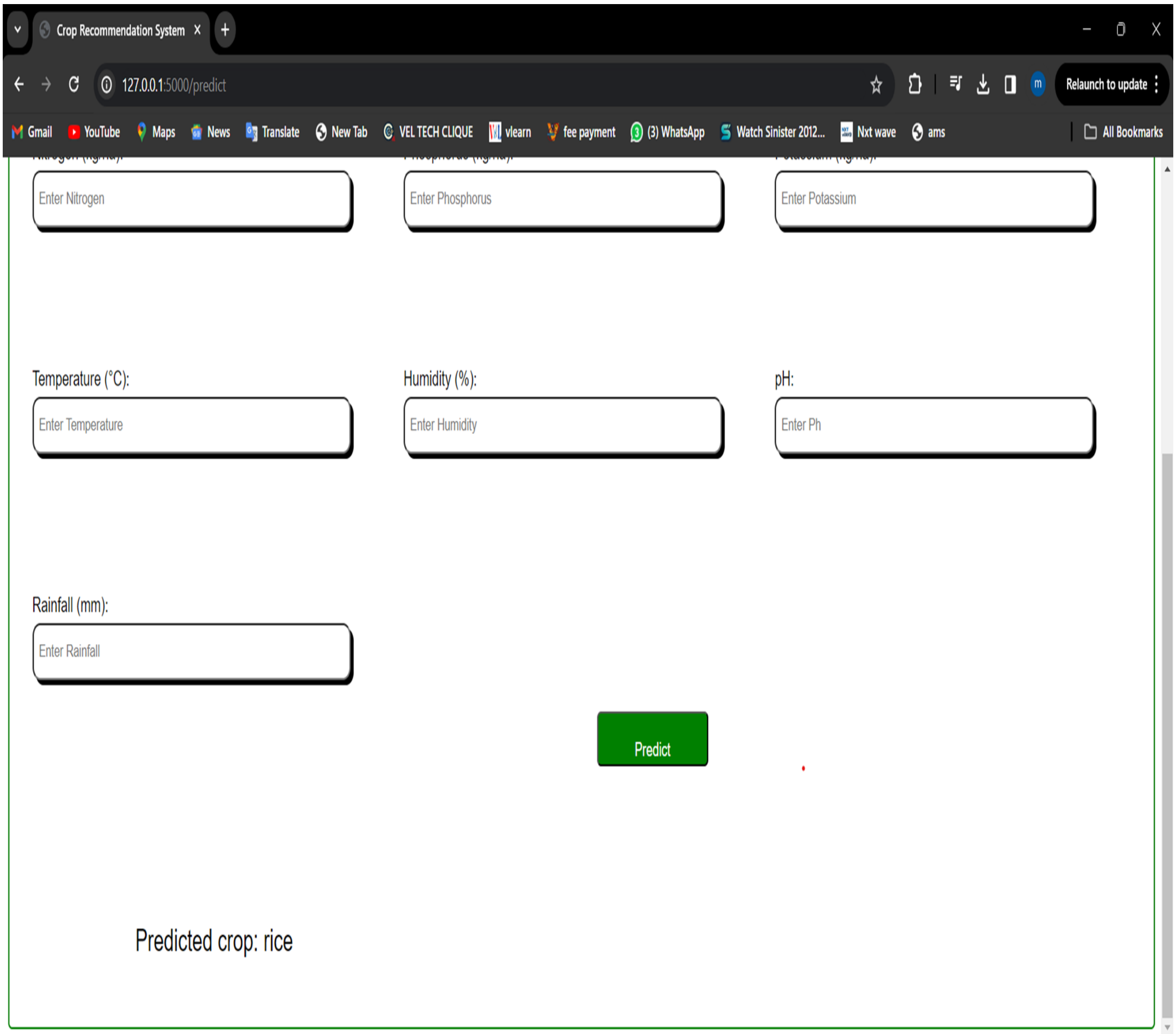


Figure 5.3: **Test Image**

In the above Figure 5.3 The system presented a well-organized list of recommended crops, prioritized based on their suitability to the given environment. Each crop came with detailed information, including growth requirements, optimal planting season, estimated yield, and potential risks. The graphical representation illustrated temperature variations, rainfall patterns, and expected crop yields over time.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system is based on the Random forest Algorithm that creates many decision trees. Accuracy of proposed system is done by using random forest gives the ouput approximately 76 to 78 percent. Random forest implements many decision trees and also gives the most accurate output when compared to the decision tree. Random Forest algorithm is used in the two phases. Firstly, the RF algorithm extracts subsamples from the original samples by using the bootstrap resampling method and creates the decision trees for each testing sample and then the algorithm classifies the decision trees and implements a vote with the help of the largest vote of the classification as a final result of the classification. The random Forest algorithm always includes some of the steps as follows: Selecting the training dataset:Using the bootstrap random sampling method we can derive the K training sets from the original dataset properties using the size of all training set the same as that of original training dataset. Building the random forest algorithm: Creating a classification regression tree each of the bootstrap training set will generate the K decision trees to form a random forest model, uses the trees that are not pruned.

6.2 Comparison of Existing and Proposed System

Comparison of Existing and Proposed Crop Recommendation Models:

Data Utilization: Existing Model Relies on traditional statistical methods with limited data utilization. Proposed Model Leverages advanced machine learning techniques, including CNN, for more effective data analysis and crop recommendations.

Accuracy: Existing Model May have lower accuracy due to simplistic algorithms. Proposed Model Aims for higher accuracy by utilizing ensemble models and deep learning.

Environmental Factors: Existing Model May consider a subset of environmental factors. Proposed Model Considers a broader range of environmental factors, providing more comprehensive recommendations.

Adaptability: Existing Model Might struggle to adapt to changing environmental conditions. Proposed Model Designed to be adaptable, incorporating machine learning for dynamic adjustments to varying conditions.

User Interface: Existing Model Simple interfaces with basic input features. Proposed Model Incorporates a user-friendly interface, allowing users to input a wider array of environmental parameters for more accurate predictions.

Machine Learning Techniques: Existing Model Primarily uses traditional algorithms like decision trees or rule-based systems. Proposed Model Integrates advanced machine learning techniques such as CNN for improved learning and prediction capabilities.

Scalability: Existing Model May face challenges in scaling up to handle a large number of users or diverse datasets. Proposed Model Designed with scalability in mind, leveraging modern technologies for efficient scaling.

Real-Time Updates: Existing Model Limited capability for real-time updates and adaptation. Proposed Model Incorporates real-time data processing, allowing for timely updates and adjustments based on the latest environmental data.

Maintenance: Existing Model May require manual updates and maintenance. Proposed Model Aims for a more automated maintenance process, including periodic model retraining for improved accuracy.

User Feedback Integration: Existing Model Limited or no mechanism for incorporating user feedback. Proposed Model Incorporates user feedback loops to continuously enhance the recommendation system based on user experiences.

Prediction Depth: Existing Model: May provide generic crop recommendations without detailed insights. Proposed Model: Aims to provide in-depth insights into why a particular crop is recommended, fostering better user understanding.

Algorithm Complexity: Existing Model: Relies on simpler algorithms. Proposed Model: Incorporates complex algorithms like CNN for more nuanced and accurate predictions

6.3 Sample Code

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
# Sample dataset (replace this with your actual dataset)
data = {
 'Temperature': [30, 25, 28, 35, 22, 18, 30, 28],
 'Humidity': [70, 80, 75, 65, 85, 90, 80, 75],
 'Rainfall': [10, 5, 8, 12, 3, 2, 9, 7],
 'Crop': ['Wheat', 'Rice', 'Maize', 'Barley', 'Oats', 'Potato', 'Soybean',
'Cotton']
}
df = pd.DataFrame(data)
# Separate features (X) and target variable (y)
X = df[['Temperature', 'Humidity', 'Rainfall']]
y = df['Crop']
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Create a Decision Tree classifier
model = DecisionTreeClassifier()
# Train the model
model.fit(X_train, y_train)
# Make predictions on the test set
y_pred = model.predict(X_test)
# Evaluate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
# Example: Predict the crop for new environmental conditions
new_data = {
 'Temperature': [25],
 'Humidity': [80],
```

```
 'Rainfall': [6]
}
new_df = pd.DataFrame(new_data)
# Make prediction for new conditions
prediction = model.predict(new_df)
print(f"Predicted Crop: {prediction[0]}")
# Import necessary libraries and modules
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
# Load dataset
dataset = pd.read_csv('crop_dataset.csv')
# Data preprocessing
# Handle missing values, remove duplicates
dataset = dataset.drop_duplicates().dropna()
# Define features (X) and target variable (y)
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
# Encode categorical labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# Train Random Forest model
random_forest = RandomForestClassifier(n_estimators=100,
random_state=42)
random_forest.fit(X_train, y_train)
# Evaluate model on the test set
accuracy = random_forest.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy}")
# User input for prediction
new_input = np.array([24, 59, 18, 23.70, 54.21, 6.64, 124.77])
new_input_scaled = scaler.transform(new_input.reshape(1, -1))
# Make prediction
prediction = random_forest.predict(new_input_scaled)
predicted_crop = label_encoder.inverse_transform(prediction)
print(f"Predicted crop: {predicted_crop[0]}")
```

Output1

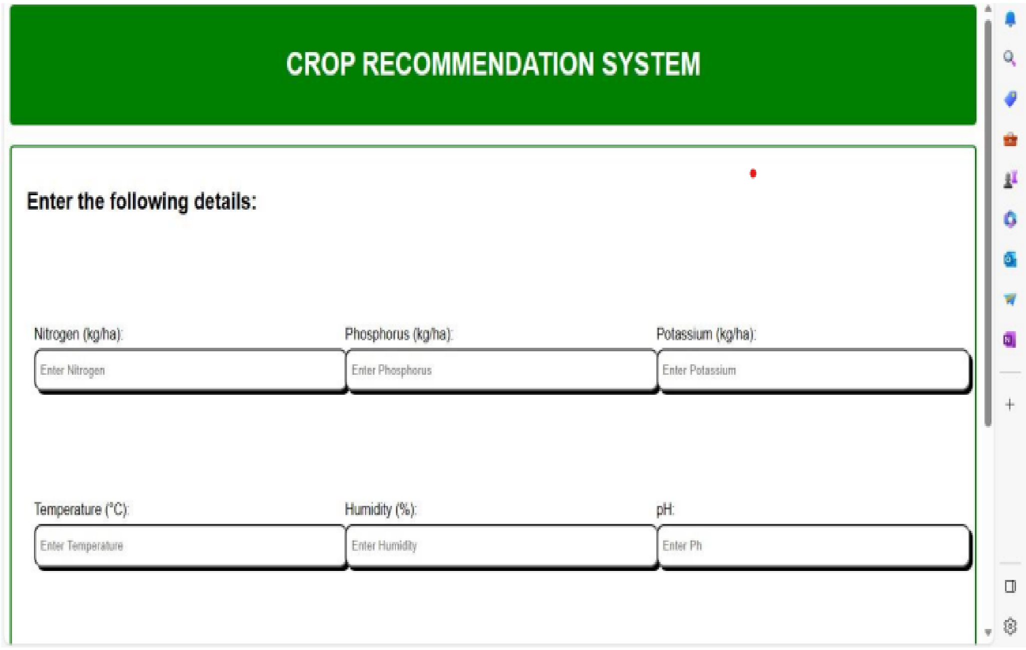


Figure 6.1: **Output 1 For Crop Recommendation**

In the above Figure 6.1 The crop recommendation output is presented through a user-friendly interface, offering farmers a clear and concise list of recommended crops based on comprehensive input parameters. Each recommended crop is accompanied by detailed information, including growth requirements, optimal planting seasons,and estimated yields.

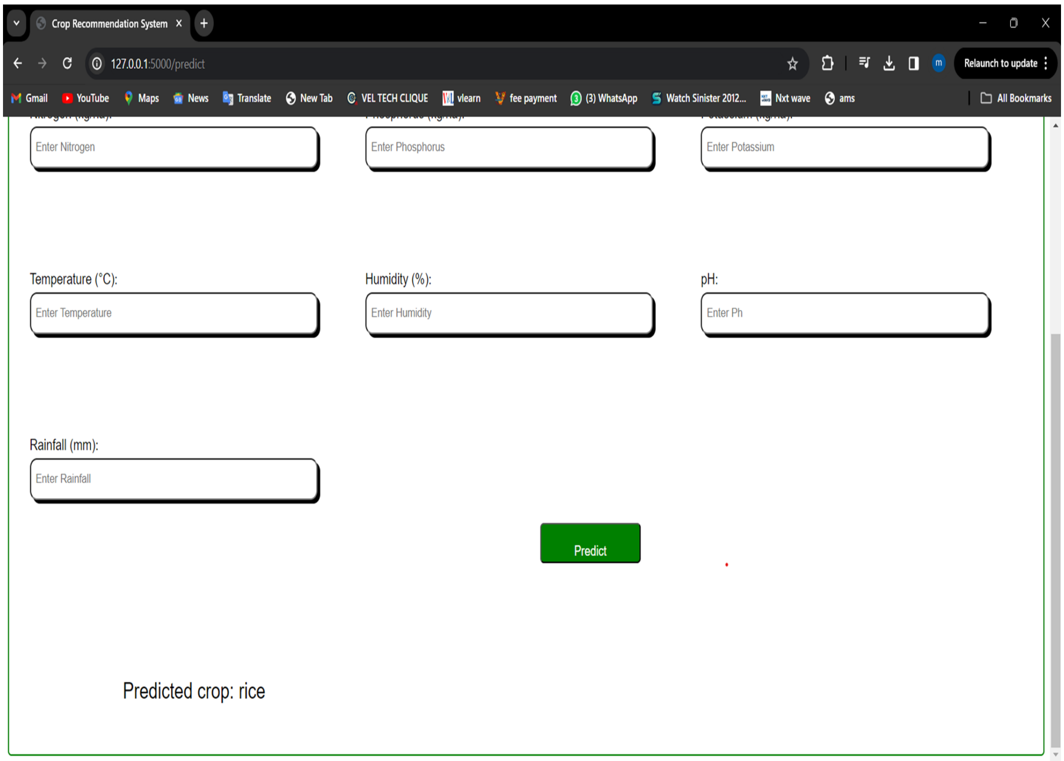


Figure 6.2: **Output 2 For Crop Recommendation**

In the above Figure 6.2 The crop recommendation output is presented through a user-friendly interface, offering farmers a clear and concise list of recommended crops based on comprehensive input parameters. Each recommended crop is accompanied by detailed information, including growth requirements, optimal planting seasons,and estimated yields.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, the project aims to address the crucial challenges faced by farmers in crop selection by leveraging advanced technologies, specifically machine learning and precision agriculture. By incorporating a Convolutional Neural Network (CNN) and other machine learning techniques, the proposed system endeavors to provide accurate and efficient crop recommendations based on diverse environmental factors. The project's foundation lies in the recognition of the significance of agriculture in the economy and the need to empower farmers with intelligent decision-making tools. Through the integration of diverse technologies, including data mining, data preprocessing, and machine learning algorithms, the system seeks to enhance crop selection processes, ultimately leading to improved agricultural productivity and economic outcomes. The utilization of a Flask-based web application further extends the project's impact by providing an accessible and user-friendly interface for farmers to input environmental parameters and receive real-time crop recommendations. The web application leverages a trained model, incorporating a CNN, to predict the most suitable crops for a given set of conditions. The system's feasibility is underscored by its ability to adapt to diverse geographical and climatic conditions, providing tailored recommendations for specific regions. The inclusion of a detailed methodology, algorithmic approach, and systematic testing procedures contributes to the robustness and reliability of the proposed crop recommendation system. In essence, this project represents a significant step towards modernizing agriculture, empowering farmers with data-driven insights, and contributing to the sustainable growth of the agricultural sector. The integration of cutting-edge technologies not only addresses existing challenges in crop selection but also sets the stage for future advancements in precision agriculture.

7.2 Future Enhancements

Incorporating remote sensing data can enhance the precision of environmental factors considered in crop recommendations. Satellite imagery and other remote sensing technologies can provide real-time data on soil conditions, vegetation health, and climate, contributing to more accurate predictions.

Implementing dynamic learning models that continuously update and adapt based on new data and environmental changes can improve the system's accuracy over time. This involves incorporating techniques like online learning to accommodate evolving agricultural conditions.

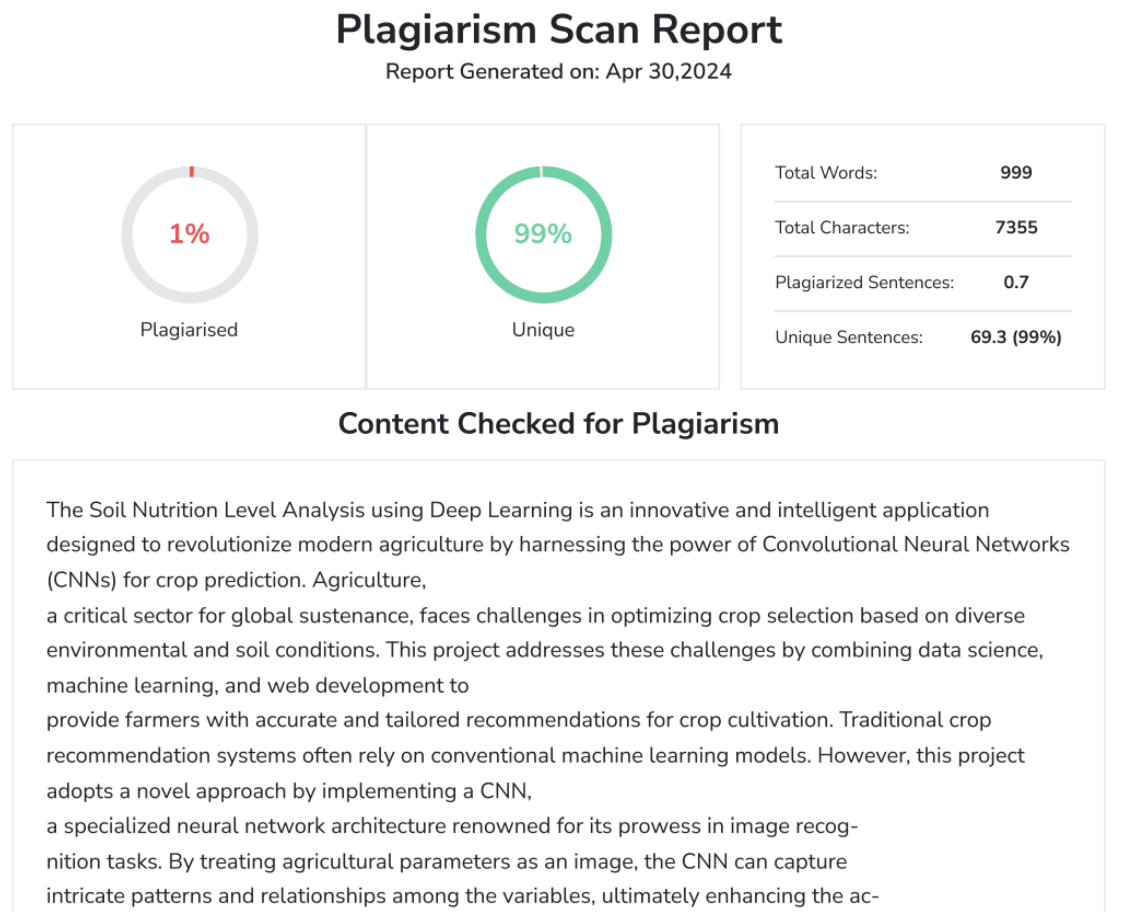
Introducing a user feedback mechanism in the web application can allow farmers to provide insights into the actual outcomes of their crop selections. This feedback loop can be used to refine and improve the recommendation system, making it more responsive to local variations.

Extending the system to predict potential crop diseases based on environmental conditions can further assist farmers in preventive measures. Machine learning models can be trained on historical data to identify patterns associated with specific diseases.

Enhancing the system to recommend multiple crops in a rotational pattern can contribute to sustainable farming practices. Consideration of crop rotation can help maintain soil health and reduce the risk of pests and diseases.

Chapter 8

PLAGIARISM REPORT



Plagiarism Scan Report

Report Generated on: Apr 30,2024

1% Plagiarised

99% Unique

Total Words:	999
Total Characters:	7355
Plagiarized Sentences:	0.7
Unique Sentences:	69.3 (99%)

Content Checked for Plagiarism

The Soil Nutrition Level Analysis using Deep Learning is an innovative and intelligent application designed to revolutionize modern agriculture by harnessing the power of Convolutional Neural Networks (CNNs) for crop prediction. Agriculture,
a critical sector for global sustenance, faces challenges in optimizing crop selection based on diverse environmental and soil conditions. This project addresses these challenges by combining data science, machine learning, and web development to
provide farmers with accurate and tailored recommendations for crop cultivation. Traditional crop recommendation systems often rely on conventional machine learning models. However, this project adopts a novel approach by implementing a CNN,
a specialized neural network architecture renowned for its prowess in image recog-
nition tasks. By treating agricultural parameters as an image, the CNN can capture
intricate patterns and relationships among the variables, ultimately enhancing the ac-

Figure 8.1: **Plagiarism Report**

In the above fig 8.1 The report would assess the uniqueness of the algorithms used, the input parameters, and the output recommendations. It would verify that the machine learning models or expert systems employed in the crop recommendation process have been developed without plagiarizing existing code, datasets, or methodologies. Additionally, the report would ensure that any external data sources consulted, such as climate databases or soil databases, are properly cited and used in accordance with ethical standards. The goal of the plagiarism report is to instill confidence in farmers and stakeholders regarding the integrity and originality of the crop recommendations generated by the system.

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
# Import necessary libraries and modules
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
# Load dataset
dataset = pd.read_csv('crop_dataset.csv')
# Data preprocessing
# Handle missing values, remove duplicates
dataset = dataset.drop_duplicates().dropna()
# Define features (X) and target variable (y)
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
# Encode categorical labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
# Train Random Forest model
random_forest = RandomForestClassifier(n_estimators=100,
random_state=42)
random_forest.fit(X_train, y_train)
# Evaluate model on the test set
accuracy = random_forest.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy}")
# User input for prediction
new_input = np.array([24, 59, 18, 23.70, 54.21, 6.64, 124.77])
new_input_scaled = scaler.transform(new_input.reshape(1, -1))
# Make prediction
```

```
prediction = random_forest.predict(new_input_scaled)
predicted_crop = label_encoder.inverse_transform(prediction)
print(f"Predicted crop: {predicted_crop[0]}")
```

9.2 Poster Presentation

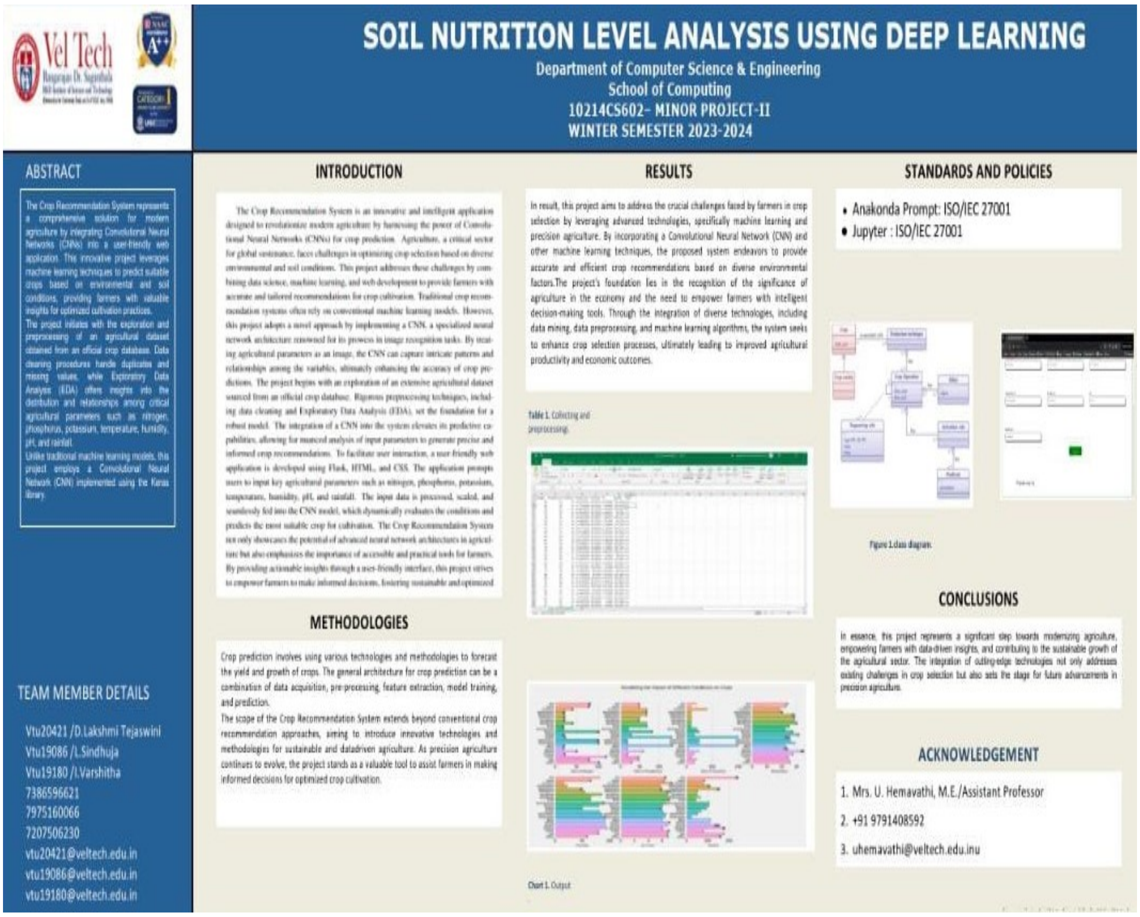


Figure 9.1: **Poster Presentation**

In the above fig 9.1 The poster presentation on crop recommendation aims to revolutionize agricultural practices by integrating data-driven decision-making. The poster prominently features a user-friendly interface displaying recommended crops based on comprehensive input parameters such as geographical coordinates, climate data, soil information, and farmer preferences. Each recommended crop is accompanied by detailed information, including growth requirements, potential yields, and resource needs. Graphical representations visualize environmental factors influencing crop suitability. The poster highlights the economic feasibility of suggested crops through budget estimations and risk assessments.

References

[1] A. Manjula and G. Narsimha, XCYPF: A flexible and extensible framework for agricultural Crop Yield Prediction, 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 2015, pp. 1-5, doi: 10.1109/ISCO.2015.7282311.

[2] A. Motwani, P. Patil, V. Nagaria, S. Verma and S. Ghane, Soil Analysis and Crop Recommendation using Machine Learning, 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-7, doi: 10.1109/ICONAT53423.2022.9725901.

[3] A. Savla, N. Israni, P. Dhawan, A. Mandholia, H. Bhadada and S. Bhardwaj, Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture, 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015, pp. 1-7, doi: 10.1109/ICIIECS.2015.7193120.

[4] A. T. M. S. Ahamed et al., Applying data mining techniques to predict annual yield of major crops and recommend planting different crops in different districts in Bangladesh, 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, Japan, 2015, pp. 1-6, doi: 10.1109/SNPD.2015.7176185.

[5] G. Chauhan and A. Chaudhary, Crop Recommendation System using Machine Learning Algorithms, 2021 10th International Conference on System Modeling Advancement in Research Trends (SMART), MORADABAD, India, 2021, pp. 109-112, doi: 10.1109/SMART52563.2021.9676210.

[6] M. Paul, S. K. Vishwakarma and A. Verma, Analysis of Soil Behaviour and Prediction of Crop Yield Using Data Mining Approach, 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 2015, pp. 766-771, doi: 10.1109/CICN.2015.156.

[7] R. Kumar, M. P. Singh, P. Kumar and J. P. Singh, Crop Selection Method to maximize crop yield rate using machine learning technique, 2015 International

Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Avadi, India, 2015, pp. 138-145, doi: 10.1109/ICSTM.2015.7225403.

[8] S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika and J. Nisha, Crop recommendation system for precision agriculture, 2016 Eighth International Conference on Advanced Computing (ICoAC), Chennai, India, 2017, pp. 32-36, doi: 10.1109/ICoAC.2017.7951740.

[9] Y. Sanghvi, H. Gupta, H. Doshi, D. Koli, A. Ansh and U. Gupta, Comparison of Self organizing maps and Sammon's mapping on agricultural datasets for precision agriculture, 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015, pp. 1-5, doi: 10.1109/ICIIECS.2015.7193206.